CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

NL 339 (r. 85/09)

National Library of Canada
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada

77

TC –

ISBN 0-315-21272-1

CANADIAN THESES ON MICROFICHE SERVICE – SERVICE DES THÈSES CANADIENNES SUR MICROFICHE

PERMISION TO MICROFILM – AUTORISATION DE MICROFILMER

- Please print or type – Écrire en lettres moulées ou dactylographier

AUTHOR – AUTEUR

Full Name of Author – Nom complet de l'auteur: Ann Helen Prystajecky

Date of Birth – Date de naissance December 25, 1961	Canadian Citizen – Citoyen canadien ☑ Yes / Oui ☐ No / Non
Country of Birth – Lieu de naissance Canada	Permanent Address – Résidence fixe 12608 51 A Avenue Edmonton, Alberta T6H 0N5

THESIS – THÈSE

Title of Thesis – Titre de la thèse: A Study of the 16B9Q Line Code for Multimode Optical Fiber Communications

Degree for which thesis was presented Grade pour lequel cette thèse fut présentée Master of Science	Year this degree conferred Année d'obtention de ce grade 1985
University – Université University of Alberta	Name of Supervisor – Nom du directeur de thèse Dr. P.A. Goud & Dr. C.G. Englefield

AUTHORIZATION – AUTORISATION

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to **microfilm** this thesis and to **lend** or **sell copies of the film.**

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de **microfilmer** cette thèse et de **prêter** ou de **vendre des exemplaires du film.**

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

ATTACH FORM TO THESIS – VEUILLEZ JOINDRE CE FORMULAIRE À LA THÈSE

Signature Ann Prystajecky	Date April 22, 1985

NL-91 (r. 84/03)

Canada

THE UNIVERSITY OF ALBERTA

A STUDY OF THE 16B9Q LINE CODE FOR MULTIMODE OPTICAL FIBER COMMUNICATIONS

by

©

ANN HELEN PRYSTAJECKY

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

ELECTRICAL ENGINEERING

EDMONTON, ALBERTA
SPRING 1985

THE UNIVERSITY OF ALBERTA
RELEASE FORM

NAME OF AUTHOR ANN HELEN PRYSTAJECKY

TITLE OF THESIS A STUDY OF THE 16B9Q LINE CODE FOR MULTIMODE OPTICAL FIBER COMMUNICATIONS

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1985

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Ann Prystajecky

PERMANENT ADDRESS:

12608 51 A Avenue
Edmonton, Alberta, Canada
T6H 0N5

DATED April 22 1985

THE UNIVERSITY OF ALBERTA
FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A STUDY OF THE 16B9Q LINE CODE FOR MULTIMODE OPTICAL FIBER COMMUNICATIONS submitted by ANN HELEN PRYSTAJECKY in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE.

Paul A. Goud........................
Supervisor

........C. G. Englefield.........
Supervisor

........R. E. Rink...............

........Wo-Sang Young.........

Date....17th April 1985

ABSTRACT

This thesis examines the applicability of the 16B9Q line code to a multimode optical fiber communication system. The necessity for line coding is first discussed. This leads to a description of the characteristics which a good line code should possess, with special emphasis on the transmission impairments of optical fiber systems that must be compensated for by the line code. Several types of line code are examined with these requirements in mind, and from them a new code classification, the unconventional multilevel block code, is defined. The 16B9Q line code, first developed and considered at Bell-Northern Research, falls into this new category. This line code maintains good code properties by selectively inverting the input data words before multilevel conversion and transmission. The coding rules, a possible framing strategy, and a method of decoding are described in detail, followed by a description of the properties of the code itself. The 16B9Q line code is then modelled as a finite state sequential machine, from which the statistics of the coding process are obtained. Following the algorithm of Cariolaro and Tronca [51] , these statistics are used to derive an analytical expression for the power spectrum of the 16B9Q line code. The derivations of the code statistics and the power spectrum are performed with the aid of several computer programs. In order to verify the code properties, an experimental system has been designed and constructed. Novel methods, described in this thesis, are used in the design of the coder and decoder, which enable these circuits to operate at bit rates exceeding 100 Mb/s. The experimental system has been tested at the DS-3 bit rate of 44.736 Mb/s, and it has been verified that use of this line code allows successful transmission over dispersion limited fiber, that the power spectrum is of the theoretically predicted shape, that the code is bit sequence independent, and that it contains enough timing information for clock extraction to be implemented readily. A series of BER tests have shown that the entire system behaves as expected. It is therefore concluded that the 16B9Q line code is eminently suitable for multimode optical fiber communication systems.

ACKNOWLEDGEMENTS

I would like to thank Dr. C. G. Englefield and Dr. P. A. Goud for their help, advice, and encouragement throughout the course of this project. Thanks and appreciation are extended to Mr. B. Telder for his technical assistance and helpful suggestions, and to the members of the examining committee for reviewing this work. As well, appreciation is extended to the following:

- Mr. A. Graves (Bell-Northern Research) and Mr. J. Dilley (Bell-Northern Research) for suggestions and discussions relating to the line code studied in this project.
- Mr. J. Slevinsky (Bell-Northern Research) and Mr. G. Ries (Bell-Northern Research) for helpful suggestions and discussions.
- Dr. E. Thompson (Northern Telecom) for supplying the optical fiber.
- Mr. L. Zalzalah (Edmonton Telephones) and Mr. D. Sloan (Edmonton Telephones) for the loan of an OTDR, fusion splicer, and connector tools.
- Mr. R. Cowper (Bell-Northern Research) for the loan of a fusion splicer, as well as helpful discussions concerning the definition of the project.
- The members of the Optical and Microwave Communications group.
- The National Science and Engineering Research Council of Canada for a post-graduate scholarship.
- The Department of Electrical Engineering for teaching assistantships.

I would also like to express my thanks to my parents, friends, and teachers who have made it possible to complete the work reported in this thesis.

Table of Contents

Chapter Page

1. INTRODUCTION 1

1.1 A Brief History of Optical Communications 1

1.2 Information Transmission 3

1.3 Characteristics of Line Codes 10

1.3.1 Code Radix 10

1.3.2 Energy Spectrum 11

1.3.3 Timing Content 12

1.3.4 Bit Sequence Independence (Transparency) 15

1.3.5 Unique Decodability 16

1.3.6 Redundancy 16

1.3.7 Information Capacity 16

1.3.8 Error Detection 18

1.3.9 Error Extension 18

1.3.10 Word Alignment 19

1.3.11 Source Nonlinearity 19

1.3.12 Average Transmitted Power 20

1.3.13 Receiver Sensitivity 20

1.3.14 Dispersion Effects 21

1.3.15 Circuit Complexity 21

1.4 Types of Line Codes 21

1.4.1 Simple Codes 21

1.4.2 Multilevel Codes 23

1.4.3 Block Alphabetic Codes 23

1.4.4 A New Code Classification 26

1.5 Thesis Objectives and Organization 29

1.5.1 Objectives 29

1.5.2 Organization 30

2. THE 16B9Q LINE CODE 31

2.1 An Unconventional Multilevel Block Code 31

2.2 16B9Q - A 2nB/(n+1)Q Line Code 32

2.3 16B9Q Coding Rules 34

- 2.3.1 Monoblock Code 34
- 2.3.2 Split Block Code 39

2.4 Framing and Decoding 42

2.5 Properties of the 16B9Q Code 45

- 2.5.1 Radix 45
- 2.5.2 Spectral Shape 45
- 2.5.3 Timing Content 45
- 2.5.4 Transparency 46
- 2.5.5 Unique Decodability 46
- 2.5.6 Redundancy 46
- 2.5.7 Information Capacity 46
- 2.5.8 Error Detection 46
- 2.5.9 Error Extension 47
- 2.5.10 Framing 47
- 2.5.11 Source Nonlinearity 49
- 2.5.12 Average Transmitted Power 49
- 2.5.13 Receiver Sensitivity 49
- 2.5.14 Dispersion Effects 49
- 2.5.15 Circuit Complexity 49

3. DERIVATION OF THE ANALYTICAL POWER SPECTRUM 50

3.1 Markov Processes 50

3.2 Power Spectrum of a Stochastic Process 54

- 3.2.1 Some Definitions 54
  - 3.2.1.1 Stationarity 54
  - 3.2.1.2 Ergodicity 55
- 3.2.2 Power Spectrum 55

3.3 Finite State Sequential Machine Model of the 16B9Q Code 58

- 3.3.1 State Probabilities 62

3.4 Spectrum of the Coded Signal 71

3.5 Application to the 16B9Q Code 79
4. THE EXPERIMENTAL 16B9Q SYSTEM 86
4.1 Coder 86
4.2 Transmitter 102
4.3 Fiber 103
4.4 Receiver 103
4.5 Decoder 106
5. EXPERIMENTAL RESULTS 116
5.1 The Experimental System Topology 116
5.2 Coder and Decoder Operation 118
5.3 Operation over Dispersion Limited Fiber 123
5.4 Power Spectrum 127
5.5 Bit Sequence Independence 131
5.6 Timing Content 131
5.7 Bit Error Rate 132
6. SUMMARY AND CONCLUSIONS 137
APPENDIX A.
DESIGN OF THE TRANSIMPEDANCE PREAMPLIFIER 144
APPENDIX B.
DERIVATION OF SYSTEM CONSTRAINTS 153
APPENDIX C.
COMPUTER PROGRAM LISTINGS 158
BIBLIOGRAPHY 190

LIST OF TABLES

Table Page

1.1 Example of a non-uniquely decodable line code. 17

1.2 ROM size needed for various mBnB line codes. 27

2.1 Translation of binary to quaternary symbols. 36

2.2 Maximum RDS values for various monoblock 2nB/(n+1)Q line codes. 40

2.3 Bandwidth reduction and efficiency for various monoblock 2nB(n+1)Q line codes. 40

2.4 Rules for calculation of the (n+1)th symbol. 41

2.5 Maximum RDS values for the monoblock and split block versions of several 2nB/(n+1)Q line codes. 43

3.1 Code word translation table for the 16B9Q line code. 59

3.2 DSV probabilities. 64

3.3 Absolute state probability vector for the 16B9Q line code. 69

4.1 Definition of schematic symbols. 88

4.2 List of coder components. 90

4.3 Creation of the ninth symbol in the 16B9Q line code. 101

4.4 List of decoder components. 110

B.1 Attenuation limits for five line codes. 156

B.2 Dispersion limits for five line codes. 157

LIST OF FIGURES

Figure Page

1.1 Block diagram of an optical fiber transmission system. 5
1.2 Eye patterns for binary and ternary signals. 7
1.3 Block diagram of an optical receiver. 9
1.4 Power spectra of several line codes. 13
1.5 Block diagram of the timing extraction process in a communication system receiver. 14
1.6 Theoretical bit rate versus repeater spacing curves for several line codes. 24
1.7 Examples of several line codes. 28
2.1 Examples of the RDS and DSV for a four level symbol stream. 33
2.2 Relative light levels for a four level optical system. 35
2.3 Example of the monoblock coding process. 38
3.1 Example of a three state Markov process. 53
3.2 Probability distribution function of the DSV of the 16B9Q line code. 65
3.3 Flowchart of the process used to determine the transition probability matrix of the 16B9Q line code. 66
3.4 The transition probability matrix of the 16B9Q line code. 67
3.5 The absolute state probability vector $\overset{\infty}{\mathbf{p}}$ of the 16B9Q line code. 70
3.6 The $\mathbf{C}_1$ and $\mathbf{C}_2$ matrices of the 16B9Q line code. 80
3.7 The $\mathbf{R}_0$ matrix of the 16B9Q line code. 81
3.8 The numerator coefficients, N_k, of the 16B9Q line code. 82
3.9 The D_k, ν_k and μ_k coefficients of the 16B9Q line code. 83
3.10 The analytically derived power spectrum of the 16B9Q line code. 85
4.1 Block diagram of the experimental 16B9Q optical fiber system. 87
4.2 Block diagram of the 16B9Q line coder. 89
4.3 The coder clock generating circuit. 91
4.4 The data acceptance, storage, and inversion circuit of the coder. 92
4.5 The coder DSV adder circuit. 93
4.6 The coder RDS adder circuit. 94
4.7 The coder ninth symbol generator circuit. 95
4.8 The binary to four level converter. 96

4.9 Experimental optical detector/preamplifier circuit diagram. 104
4.10 Expected frequency response of the transimpedance preamplifier. 105
4.11 The experimental main amplifier, peak detector, and automatic gain control circuits. 106
4.12 Block diagram of the overall 16B9Q decoder. 108
4.13 Block diagram of the implemented 16B9Q decoder. 109
4.14 The decoder clock generating circuit. 111
4.15 The four level to binary converter. 112
4.16 The decoder inversion and output circuit. 114
5.1 The experimental 16B9Q system test set up. 117
5.2 The four level 16B9Q coded signal and its constituent magnitude and polarity waveforms for an input data word of 1111 1111 1111 1111. 119
5.3 The four level 16B9Q coded signal and its constituent magnitude and polarity waveforms for an input data word of 0000 0000 0000 0000. 120
5.4 The four level 16B9Q coded signal and its constituent magnitude and polarity waveforms for an input data word of 0001 1000 0000 0000. 121
5.5 The four level 16B9Q coded signal for an input 2^{23}-1 PRBS. 122
5.6 Theoretical bit rate versus range curves for the experimental 16B9Q system utilizing highly dispersive multimode fiber. 124
5.7 Experimental results of transmitting a binary encoded signal at a bit rate of 44.736 Mb/s over a 25 MHz bandwidth multimode fiber. 125
5.8 Experimental results of transmitting a 16B9Q signal at a bit rate of 44.736 Mb/s over a 25 MHz multimode fiber. 126
5.9 Power spectra of three different 16B9Q encoded PRBS signals at the coder output. ... 128
5.10 Low frequency power spectrum of a 16B9Q encoded 2^{23}-1 PRBS. 129
5.11 Power spectrum of a 16B9Q encoded 2^{23}-1 PRBS at the output of the preamplifier. .. 129
5.12 Computer simulated power spectrum of the 16B9Q line code. 130
5.13 Effect of the variation of APD gain on the experimental system BER. 133
5.14 Effect of the position of comparator threshold level on the experimental system BER. 134
5.15 Effect of the variation of the detector input power on the experimental system BER. 136
A.1 AC circuit of the experimental preamplifier. 145

A.2 Equivalent circuit of the preamplifier. 146
A.3 Simplified preamplifier equivalent circuit. 149
B.1 Attainable receiver sensitivities for binary signalling for a 10^{-9} BER. 155

CHAPTER 1

INTRODUCTION

1.1 A Brief History of Optical Communications

The concept of using optical frequencies for information transmission is not a new one, nor one restricted to man. Examples of optical communication found in the animal kingdom include courtship displays and warnings of aggresive intent, as well as the actual use of light by the firefly and various ocean dwelling animals. Optical techniques of sending information over long distances have been employed by man from the earliest times. Although sophisticated methods of signalling were discussed by the ancient greeks, the only practical means in use over the centuries were line of sight systems such as fire beacons [1]. The end of the eighteenth century heralded the development of the semaphore, but it was not until the late 1800s when experimentation in key areas of modern optical communications began. John Tyndall, a British physicist, first brought the effect of light guiding to the public eye in 1854, by demonstrating how a light beam followed a zigzag path inside a curved stream of water, confined there by total internal refraction. William Wheeler, an American engineer, envisaged a practical use for guiding light - to distribute illumination throughout buildings. He obtained a patent for a light piping scheme (reflective pipes were used to distribute light from a single, central source throughout a building) in 1881. In 1880 Alexander Graham Bell developed the photophone. By intensity modulating a beam of sunlight, this device transmitted a telephone signal over a distance of two hundred meters. Although it did not reach commercial application (due to the lack of a reliable intense light source and a dependable low loss transmission medium), the photophone demonstrated the basic principles of optical communications as practiced today [1,2,3]. At approximately the same time, Charles Vernon Boys, a member of the physics faculty of the Royal College of Science in South Kensington, experimented with the drawing of glass fibers. However, the demonstration of the existence of radio waves by Heinrich Hertz in 1884, and the successful radio transmission experiments of Guglielmo Marconi (starting in 1895 and culminating with a transatlantic transmission in 1901), captured the imagination of many scientists. Communications by way of light waves thus faded into the background. German engineers and scientists, though, remained interested in optical signalling. During World War I, a military communication system was built, which used a voice modulated carbon arc and had a range of 8 kilometers. German, Italian, Japanese and American forces in World War II used various types of optical communication systems, some with ranges exceeding 10 kilometers. All were bulky, however, and most relied on the mechanical modulation of light. By comparison, radio communication systems were much more advanced and could be used in conditions where light was blocked. Thus, for the next several years, radio developed at a faster rate than optical systems.

The invention of the laser in 1960 marked the beginning of the era of modern optical communications. Highly monochromatic, coherent, and intense laser radiation can be used for unguided propagation through the atmosphere, as well as for guided propagation through dielectric waveguides. The initial motivation for the further development of optical communications was the enormous bandwidth that would be available if it were possible to modulate a laser at a few percent of its fundamental frequency [1]. The electromagnetic spectrum in the optical region encompasses frequencies of the order of hundreds of THz. One percent of this gives a channel bandwidth of hundreds of GHz.

Extensive research began in the 1960s on optical devices, components and subsystems, as well as on various transmission media, such as waveguides with lenses, prisms and mirrors as focusing elements, waveguides filled with heated flowing gases (in order to establish a refractive index gradient), and hollow tubes with metallic or multilayer dielectric film coatings. During this period, work on dielectric waveguides was mainly theoretical, since the available glass fibers exhibited losses that were much too large for telecommunications applications (of the order of 1000 dB/km). By 1970, however, the threshold at which communications applications were considered to be feasible had been reached with the development of a doped silica fiber exhibiting an attenuation of 20 dB/km [4]. Following this achievement, progress in all areas of optical communications has been very rapid. Glass fibers with a loss as low as 0.1 dB/km have been developed, while laboratory work currently in progress on halide materials will yield fibers exhibiting an attenuation of the order of 0.001 dB/km [5]. Semiconductor lasers and light emitting diodes continue to be perfected as sources, concurrently as PIN and avalanche photodiode (APD) detectors are improved. Connectors and splicing methods, as well as cabling methods for optical fibers have also received considerable attention. Not only has work progressed in all of these areas, but first generation systems consisting of short wavelength (.8μm to .9μm) sources and detectors along with multimode fiber, have already been superceded by second generation systems using long wavelength (1.3μm and 1.55μm) sources and detectors and monomode fiber.

The development of optical fiber technology and its application to telecommunications has been incredibly rapid, with the elapsed time between laboratory research and commercial system installation encompassing less than twenty years. This speed of advance can be attributed to the many advantages optical fiber offers over other transmission media, as well as to the availability of simultaneously developing technologies such as high speed electronics. During the 1970s, an urgent necessity for channel bandwidth became apparent. To fulfill this need, millimeter microwave waveguide systems were developed and field trials were performed. However, these systems proved to be extremely expensive, and optical fiber systems were seen as a reasonable alternative. Optical fibers exhibit an extremely large bandwidth (especially single mode fibers) and a very low loss (which will become even lower as new materials are developed). The material from which fibers are produced is quite common, in good supply, and inexpensive, thus considerably reducing the cost of the fibers themselves. Optical fibers are immune to electromagnetic interference and can be used with complete safety in environments where electrical sparks pose a serious threat. Fiber systems are also very secure (practically immune to tapping without detection), small, and lightweight. The significance of these advantages can be inferred from the large number of commercial optical fiber communication systems installed world wide since 1976 [4], and from the vast amount of ongoing research.

1.2 Information Transmission

Transmission of information over long distances requires the existence of a transmission medium between the originating and receiving end points. Ideally, the signal that reaches the receiver should be identical to the one that left the transmitter. However, the intervening medium is not ideal and introduces distortion of various kinds. This distortion can degrade the signal to such an extent that the received waveform becomes meaningless. In addition to the constraints imposed by the channel, the transmitter and receiver also impose certain constraints and requirements that must be met in order for the signal to reach its destination in recognizable form. Therefore, the signal to be transmitted should be converted into a form suitable for propagation over the medium with as little distortion as possible. The process of matching the signal to the transmission system and path in order to facilitate optimal transmission is called line coding.

As shown in Figure 1.1, an optical fiber transmission system consists of three sections, each of which imposes certain requirements on the characteristics of the line code.

The optical source in the transmitter is a light emitting diode (LED) or an injection laser diode (ILD). The ILD can be modulated at a much faster rate than the LED (the maximum modulation frequency of a LED is typically in the 100 MHz range, while that of the ILD is in the GHz range) and thus is the source used in high bit rate systems. ILDs, however, have a nonlinear power versus current characteristic, which can make their use in analog and multilevel digital applications difficult. Although LEDs are fairly linear, their best performance is also achieved with on-off codes (i.e., two level codes) [6]. The dynamic range of the signal that can be transmitted is restricted by the peak power limitation of the optical sources. In addition, both types of source have a finite line width (typically 40 nm for the LED and 1 nm for the laser). Their output thus contains a range of optical wavelengths, each of which travels through the fiber at a different velocity. The different wavelengths arrive at the receiver at different times, causing an initially narrow input pulse to broaden. This phenomenon is called chromatic dispersion, and it is one of the two principal contributors to the total dispersion of the system. It can be reduced by narrowing the source line width or by operating at the wavelength of minimum dispersion (about 1.3μm for SiO_2 glasses).

The second contribution to total dispersion arises from the differing group velocities of different modes. The propagation of optical modes along the fiber is determined by the type of fiber used and is discussed in greater detail below. In addition, modal noise can also depend on the type of source used. In a laser diode, the optical radiation within the resonance cavity sets up a pattern of electric and magnetic field lines, which constitute the modes of the cavity. They can be separated into two independent sets of modes: transverse electric and transverse magnetic. Longitudinal modes are related to the length of the cavity and determine the principal structure of the frequency spectrum of the emitted radiation. Since the length of the cavity is much larger than the wavelength of the emitted radiation, many longitudinal modes can exist. Lateral modes depend on the width of the cavity and determine the shape of the lateral profile of the laser beam. Transverse modes determine the radiation pattern of the laser and the threshold current density, and are associated with the height of the laser cavity. Optical output from a laser diode can arise simultaneously from all the longitudinal modes, or it can switch

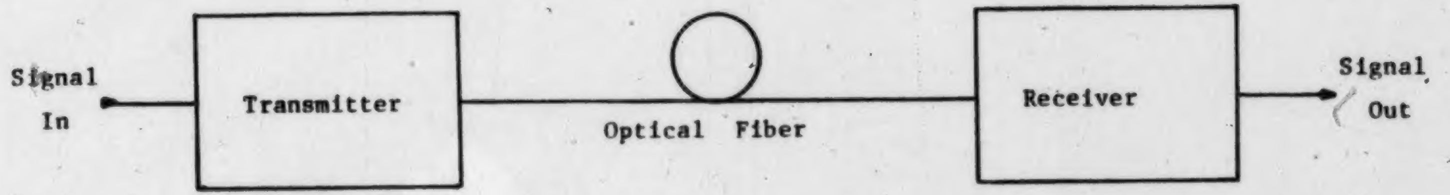
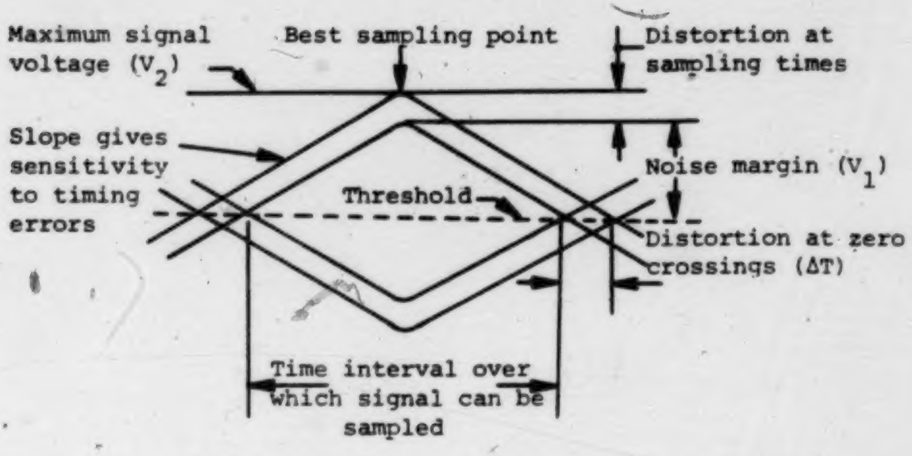


Figure 1.1. Block diagram of an optical fiber transmission system.

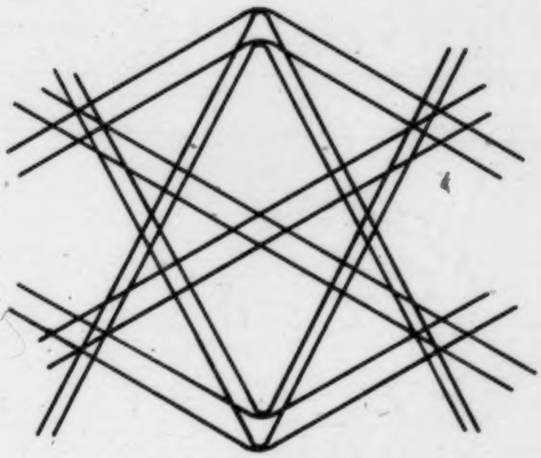
randomly from mode to mode. Even though the total output power of the laser may not vary, intensity fluctuations can exist among the various modes. These modes are coupled into the fiber with a high efficiency, and each has a different attenuation and time delay. This not only causes pulse broadening, but also results in a speckled light pattern across the end face of the output fiber, if the time delay between any two modes is shorter than the coherence time of the laser. Receiver performance is thus degraded [7]. In addition to the use of single mode lasers, a suitable line code will also help to alleviate this problem.

The transmission channel typically consists of doped silica fiber, which is a dispersive medium. Thus, a rectangular pulse coupled into the fiber at the transmitter will arrive at the receiver with a waveshape that has spread in time and is no longer rectangular. This signal distortion is due to the wavelength nature of the source as discussed above, and also to the modal nature of the fiber. It occurs in both time and frequency and is affected by external effects such as microbending. The propagation of light along a waveguide can be described by a set of guided electromagnetic waves, called the modes of the waveguide. Only a certain discrete number of modes, which satisfy the homogeneous wave equation in the fiber and the boundary condition at the waveguide surfaces, can propagate along the fiber. Each mode is a pattern of electric and magnetic field lines which is repeated along the fiber at intervals of a wavelength of the propagating light. Modal dispersion occurs because each mode travels at a different group velocity. Thus, the modes arrive at the output of the fiber at different times, resulting in pulse broadening. Modal dispersion can be reduced by grading the refractive index of the fiber in order to equalize path lengths or by using single mode fiber, which allows only one mode to propagate. The total dispersion of the system, caused by modal and chromatic dispersion, determines the bandwidth-distance product of the fiber. This figure of merit can vary from lower than 10 MHz-km to higher than 10 GHz-km, depending upon the combination of the type of fiber and source.

Pulse broadening which occurs in an optical fiber system can result in intersymbol interference, as one pulse spreads into the time slot of adjacent ones. The degree of intersymbol interference can be observed from the signal formed by the superposition of all the possible demodulated waveforms of the random data at the receiver. This signal is called an eye pattern, due to its resemblance to a human eye. Figure 1.2 illustrates eye patterns for binary and ternary signals.


Maximum signal voltage (V_2)
Best sampling point
Distortion at sampling times
Slope gives sensitivity to timing errors
Threshold
Noise margin (V_1)
Distortion at zero crossings (ΔT)
Time interval over which signal can be sampled

(a)



(b)

Figure 1.2. Eye patterns for binary and ternary signal (from Keiser [7]).

(a) Binary eye pattern.

(b) Ternary eye pattern.

The width of the eye opening defines the time over which the received signal can be sampled without error from intersymbol interference, with the optimum sampling time occurring when the height of the opening is largest. This height is reduced by amplitude distortion in the data signal. The maximum distortion is determined by the vertical distance between the top of the eye opening and the maximum signal level. As the eye closes, the signal becomes more difficult to detect, and the noise margin, which is determined by the height of the eye at the sampling time, decreases. The slope of the eye pattern sides (i.e., the rate at which the eye closes as the sampling time varies) is a measure of the sensitivity of the system to timing errors. As the slope becomes more horizontal, the possibility of timing errors increases. Timing jitter in an optical fiber system results from noise in the receiver and pulse distortion in the fiber. If the signal is sampled in the middle of the timing interval, the amount of distortion at the threshold level (the difference between threshold crossings) indicates the amount of jitter. Any nonlinearities of the channel transfer function create an asymmetrical eye pattern. A random data stream passed through a linear system will have an eye pattern with identical, symmetrical eye openings. The eye pattern can thus be used to evaluate the performance of an optical fiber system.

A code which minimizes dispersion effects, resulting in an open eye and a lower error probability, is therefore desirable.

The optical receiver is composed of a PIN photodiode or APD, followed by a preamplifier, amplifier, equalizer, and decision circuit, as shown in Figure 1.3. Since the transmission medium is a dielectric material, the receiver is electrically isolated from the transmitter. This means that the transmitted data cannot contain an information carrying DC component (as it will be lost due to AC coupling in the receiver) and must not be susceptible to improper decoding caused by DC drift. The power spectral density should therefore have a small, preferably zero, DC value. In order to properly regenerate the incoming signal the data must be sampled at appropriate instants. For a self-synchronizing system, therefore, the data itself must contain enough timing references (level transitions) for the receiver to be able to extract a clock signal to use for this purpose.

Suitable coding is therefore necessary for the efficient utilization of the transmission channel and for the successful decoding of the transmitted data.

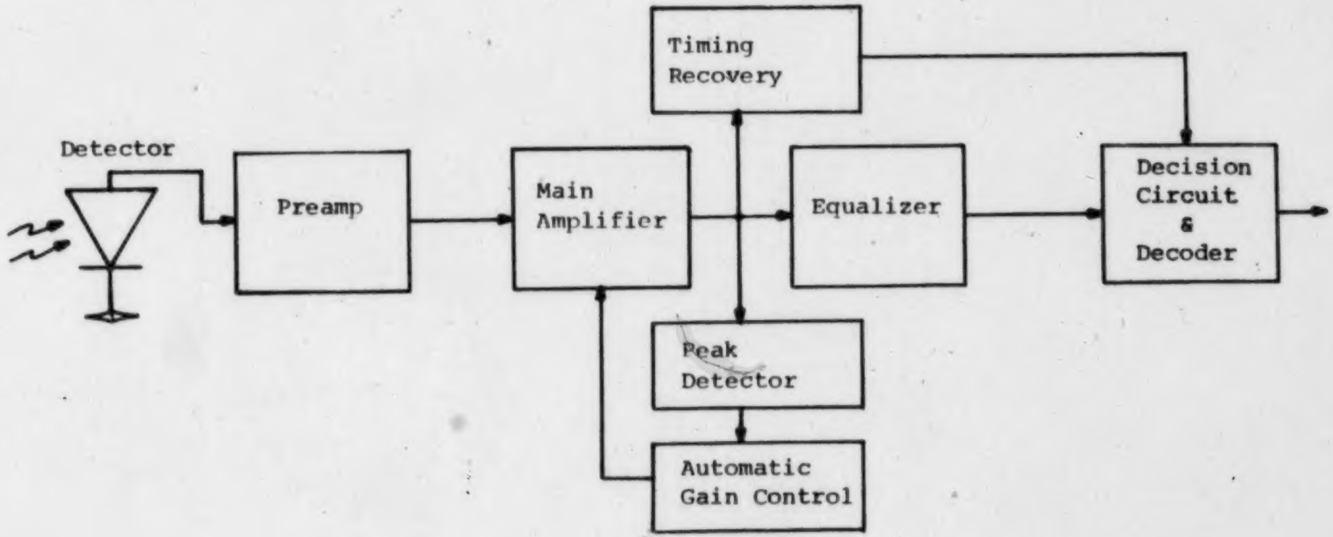


Figure 1.3. Block diagram of an optical receiver.

1.3 Characteristics of Line Codes

Various features require consideration in the process of line code selection [7-33]. A number of these are discussed below.

1.3.1 Code Radix

The radix is the number of amplitude or width levels, or pulse positions used in a code. The transmitted signal is distorted due to source nonlinearity, channel dispersion, scattering at fiber splices and connectors, jitter, threshold inaccuracy, and imperfect equalization. The number of distinguishable amplitude or width levels in the coded data stream is limited by the total amount of distortion. The greater the distortion, the fewer the number of distinguishable levels. In order to implement a specific number of levels, therefore, the distortion must be kept within certain limits. This is more easily accomplished if the radix is low, since the allowable distortion is then greater and less complex methods can be used to keep it under control. The eye openings of the demodulated signal must be wide and the data levels easily detectable at the sampling instants in order for correct decoding to occur. This can again be achieved with a code having a low radix, as multilevel eye patterns have much less threshold tolerance in both the horizontal and vertical directions [7].

First generation optical fiber systems used ILDs extensively as sources (due to their higher coupled power and resulting greater repeaterless range) and silicon APDs as detectors (due to their greater sensitivity at the operating wavelength). The fiber was multimode and fairly dispersive. These components presented conflicting requirements in the choice of a code radix. The output characteristics of the initial lasers were far from linear, thus indicating the use of a binary code. Signal dependent shot noise in the receiver required non uniformly spaced signal and decision levels in order to achieve optimal noise performance. This complicated not only the receiver, but also the transmitter drive circuits, and a binary code presented a much simpler system. A multilevel code, however, could exploit the capacity of the highly dispersive fiber much more effectively than a binary code, as much higher bit rates could then be transmitted.

Advances in laser technology have greatly improved the output characteristics of ILDs, so that they are now fairly linear and do not present an obstacle to multilevel coding. The most sensitive receivers for long wavelength systems are PIN-FET combinations. For these receivers, shot noise effects are of secondary importance and it is therefore possible to use uniformly spaced signal and decision levels. Thus, the major obstacle to the use of multilevel coding has disappeared.

The choice in code radix will be affected by complexity considerations, the type of receiver, the bit rate, the type of fiber, and the length of the system. The circuits used in implementing a multilevel code will generally be more complex than those for a binary code. In APD and PIN diode receivers, multilevel codes do not show an advantage over binary codes unless there is significant fiber dispersion. With a PIN-FET receiver, multilevel codes have a small sensitivity penalty for single mode systems, while for multimode systems they quickly show an advantage as dispersion effects become noticable [8]. Long wavelength, high capacity systems (140 Mb/s or greater) tend to be dispersion limited, due to the low fiber attenuations at these wavelengths. In this situation multilevel codes offer a significant advantage, as they can be used to increase the distance between repeaters or to increase the bit rate for a fixed repeater section. Thus, existing systems can be upgraded by simply changing the transmission code. Multilevel transmission can also be useful in low bit rate systems, which are usually short and contain no intermediate repeaters. In these cases it might be economically attractive to transmit the AMI or HDB3 interface code without any additional coding. Another possible use for multilevel codes lies in high bit rate systems (greater than 565 Mb/s) where the operating speed of the devices would consequently be reduced. In such situations it could be preferable to sustain slightly increased repeater complexity, instead of changing a proven technology. In the choice of code radix, therefore, each system must be considered on an individual basis

1.3.2 Energy Spectrum

In order to receive a transmitted signal with minimum distortion its complete frequency spectrum should be transmitted over the channel. Thus, the signal spectrum should fall within the passband of the system, requiring spectral shaping by means of a line code. The optical receiver is electrically isolated from the transmitter, and is therefore AC coupled to the fiber link. The power spectral density must therefore exhibit a null or a minimum at DC. Baseline wander, the shift of the baseline due to the average time the signal is positive relative to the average time it is negative (in optical terms positive indicates a level above the mean light level, while negative indicates a level below it) creates a low frequency component in the transmitted signal and causes eye closure. It is a desirable feature for the line code to prevent this by not allowing a long sequence of positive or negative excursions. In addition, the receiver is very sensitive to AC signal variations. The decision as to which level was transmitted is determined by positive or negative peaks of the signal at the sampling time. When a constant level is transmitted, the AC coupled preamplifier output will drift toward its center level and will not remain at the level corresponding to the information being transmitted. Consequently, the decision logic will incorrectly decode the signal. Also, a long period without transitions will cause timing problems to occur. In order to alleviate this, the spectrum should contain minimum low frequency energy. This will also permit the use of low frequency laser slope control techniques and the provision of a low frequency supervisory channel. The bandwidth of the receiver is minimized in order to decrease the effects of noise. Therefore, to prevent distortion due to filtering of high frequency components, the portion of the spectrum outside the passband should also be small. The high frequency spectrum of a pulse is important for equalization purposes and for determining the optical power penalty. The high frequency spectrum due to the code structure is generally not as important, since optical fibers do not usually suffer from crosstalk [8].

Spectral shaping also has a large impact on the timing content of the signal. A large amount of power at half the signal rate simplifies timing signal recovery (through the use of a frequency doubler). Various code structures can insert a null in the power spectrum, facilitating the insertion of a timing signal at that frequency. Finally, many codes contain a discrete as well as a continuous spectrum. The discrete lines can be filtered out and used in clock recovery schemes. Figure 1.4 compares the power spectra of several line codes.

1.3.3 Timing Content

A crucial requirement in the receiver is the identification of the point at which to sample the incoming signal sequence in order to regenerate the data. Since disturbances and distortions occur in the channel, it is important that the sampling sequence be synchronized with the transmitter clock at all times. This is accomplished by extracting clock information from the level transitions of the received signal. The timing extraction process is shown in Figure 1.5. The input signal is subjected to a nonlinear process (typically a differentiator followed by a rectifier), followed by a threshold detector and a bandpass filter (BPF) or phase

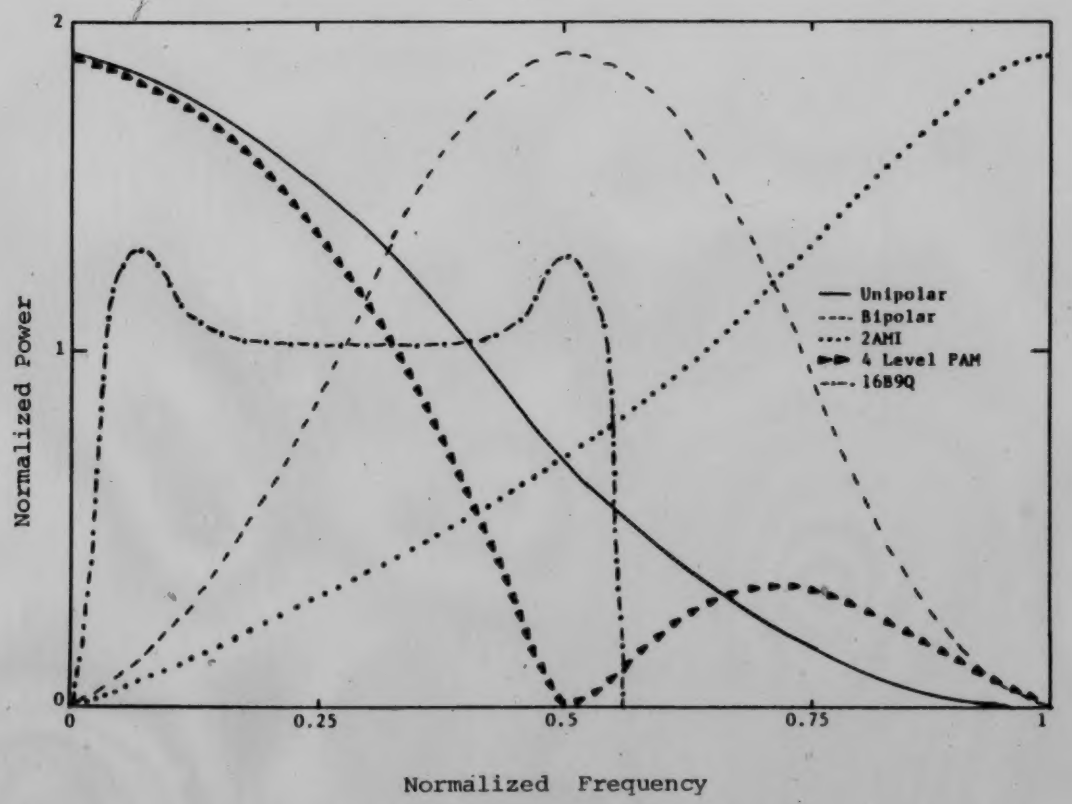


Figure 1.4. Power spectra of several line codes.

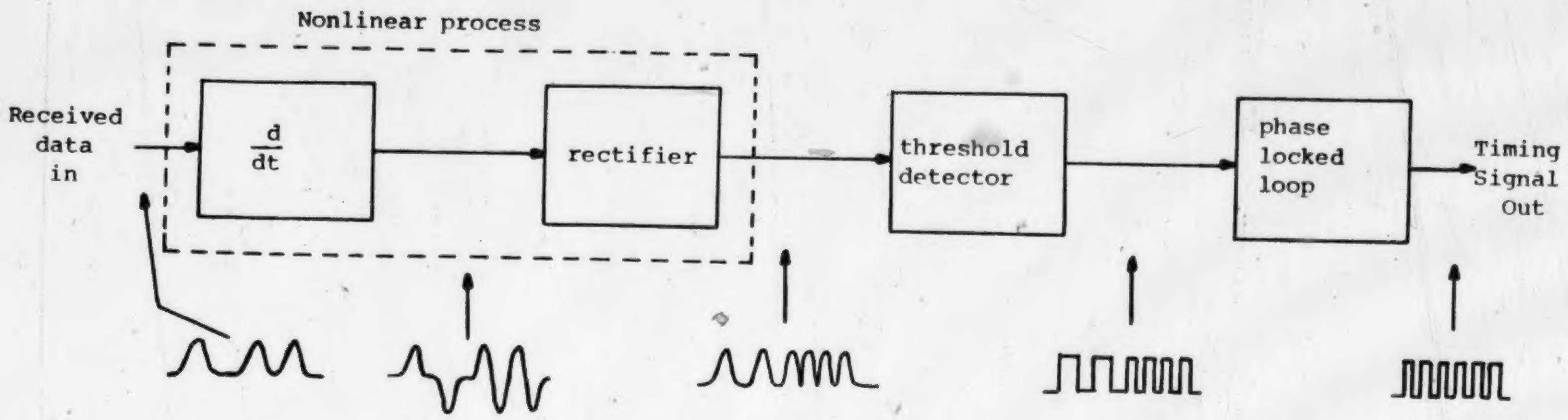


Figure 1.5. Block diagram of the timing extraction process in a communication system receiver.

locked loop (PLL). The output of the BPF or PLL provides the required timing wave at the signal frequency. Both of these circuits will continue to provide a timing signal output even if input transitions are absent for a short period of time. The allowable transitionless time is limited, however [9,10].

The timing content of a code is described in terms of the statistics of the symbol transitions. The longest interval of consecutive like elements which can occur is generally restricted. This characteristic is important not only for timing, but because long strings of like elements can cause mode hopping effects in lasers. It has also been noted that the maximum number of consecutive like elements increases with increasing code efficiency [8]. Timing extraction is much simpler if the timing information in the line signal is fairly regular. The higher the minimum density of transitions within the coded signal, the lower the effective Q needed for the timing circuit (and thus the simpler and less expensive the circuit). This gives rise to less stringent frequency stability requirements, and allows a greater number of regenerators before the jitter specification (time dependent phase shift)is exceeded. If the code does not satisfy these demands,that is, if it allows long time intervals without level transitions, a scrambler can be used to break up structure within the data and reduce systematic jitter. This, however, adds some complexity, and the possibility of error extension.

1.3.4 Bit Sequence Independence (Transparency)

The term "bit sequence independence" means that no restrictions are imposed on the content of the transmitted message. In other words, any data sequence can be presented to the coder and it will be transmitted without degradation in performance. Balanced alphabetic codes (codes which ensure a zero DC level by allowing only a specific set of codewords to be transmitted, depending upon the state of the coder) are generally constrained within acceptable limits for any source data sequence and are ,therefore, bit sequence independent. In spite of this, scrambling is used to average out the performance and improve realignment and error detection techniques. Unbalanced codes (for example, parity check codes on which there are no codeword restrictions) are generally not bit sequence independent. For these codes, a possibility exists that certain source data sequences will cause a catastrophic performance degradation. Scrambling is used here to randomize the input data, and performance is kept within acceptable limits on a statistical basis [8].

1.3.5 Unique Decodability

Although a code might give a unique output for each input combination, this does not guarantee that the output can be unambiguously decoded into the original sequence of inputs. The condition that must be satisfied in order to ensure unique decodability is that no code word may form the first part, or prefix, of any other code word. This can be seen from the code shown in Table 1.1. The third and fourth symbols of this code cannot be uniquely decoded, since those sequences could have resulted from combinations of the first two symbols. However, a line code can still use two or more different output words for a given input, or the same output for two different inputs, provided that the rest of the transmission contains information that makes it possible to decide what was sent. This leads to the stronger property of instant decodability. Many codes are such that any incoming code word can be decoded as soon as its reception is complete. That is, all the necessary information is contained in the code word itself. Although this property is not essential, it is quite useful.

1.3.6 Redundancy

The redundancy of a code is a measure of its unused capacity. The total capacity of the code is not used to transmit message information, and it is this unused portion which allows us to introduce various desirable features (such as adequate timing content and error detection capability) into the line code. The greater the portion of available capacity allocated for this purpose, the easier it is to include such features . However, the code becomes less efficient, requiring an increasing system bandwidth. In optical systems, the lower the efficiency, the larger the effect of dispersion becomes [8]. Thus, a good line code maximizes efficiency, while providing enough redundancy to implement desirable features.

1.3.7 Information Capacity

To transmit a maximum amount of information in a minimum amount of time, the number of information bits represented by each coded symbol should be as high as possible. This requirement conflicts with the need for redundancy. As the information capacity increases, the amount of redundancy decreases, and a compromise is necessary.

Table 1.1. Example of a Non-Uniquely Decodable Line Code

Binary Input	Coded Output
00	1
01	10
10	101
11	110

1.3.8 Error Detection

The performance of a digital communication system is determined primarily by the continuous monitoring of its error rate. It is necessary, therefore, to have the ability to detect errors without interruption of service and without assuming anything about the signal being carried. Decoded errors are related to transmission errors, by taking into account possible error multiplication effects of the decoding and descrambling processes. This relationship is a statistical one; therefore only an approximate measure of bit error rate (BER) is obtainable from transmission errors. An error detection capability can be obtained by providing an appropriate redundancy in the line code. Error correction of the received signal is generally not a concern in telecommunication systems, unless "real-time" correct data are needed. Otherwise, it is often more economical to retransmit data that has been received in error, than to employ error correcting codes. Error correcting codes do not in general contain the transmission properties discussed in this section and thus generally do not satisfy the requirements of line coding.

Error detection is based upon the identification of the violation of code properties such as invalid words and sequences. If the code does not include such properties, parity checks can be added, although they will increase both the symbol rate and the redundancy.

In addition to overall system performance monitoring at the receiver, error detection circuits at intermediate repeaters provide a check on the functioning of the repeater and give advance warning of impending faults. Thus, problems can be isolated to particular repeaters or portions of transmission line.

1.3.9 Error Extension

One error in the coded signal may give rise to a large number of errors in the decoded binary bit stream due to symbol interactions in the decoding operations at the receiver. This phenomenon is called error multiplication, or error extension. Error multiplication occurs in alphabetic codes and codes which use synchronizing scramblers. The error multiplication factor can be defined as either the mean or maximum number of output errors for one transmission error, while error spread is either the mean or maximum number of output digits over which the errors are spread. Although error multiplication for alphabetic codes depends on the specific code tables used, approximate values for these parameters are given by error multiplication factor = m/2

maximum error spread < m

where m is the length of the input binary word [8]. Both of these factors should be minimized in the code design.

1.3.10 Word Alignment

If the coding scheme is based upon blocks of transmitted digits, the decoder must be able to group the received symbols into the correct blocks, in order to properly decode them. In other words, the symbols have to be aligned or framed, and the alignment information must be obtained purely from the structure of the incoming serial symbol sequences. Alignment methods include the recognition of the constraints and correlations of the code conversion rules, and the detection of extra patterns ,if any, inserted into the coded symbol sequence. Following system power up, the initial alignment time is not important. The time needed to regain alignment if it is lost during normal transmission is, however, a major consideration, and it is desirable that this time be as short as possible. In general, the larger the code blocks, the longer the realignment time. It is also important that the criteria used to determine loss of frame are such that they do not cause spurious reframe. Spurious reframe occurs when the decoder decides that it is out of frame due to the large number of errors in the decoded bit stream, but in reality it is in frame and the errors are due to transmission errors. Thus, the reframe process is initiated when there is no need for it, causing a certain amount of incoming data to be lost. The time taken to detect an out of frame situation should therefore be long enough to prevent spurious realignment, but short enough to give an acceptable realignment time.

1.3.11 Source Nonlinearity

Nonlinear distortions occur in the optical output power versus current characteristics of both LEDs and ILDs. In LEDs they are caused by effects determined by the carrier injection level, radiative recombination, and other subsidiary mechanisms [7]. In ILDs these nonlinearities result from inhomogeneities in the active region of the device and from power switching between the dominant modes of the laser. Power saturation can also occur due to the heating of the active layer at high output power levels. As a result of source nonlinearities much effort has been directed to operating optical transmitters in a binary digital mode, which is fairly insensitive to these distortions [13]. However, high radiance Burrus LEDs and single mode laser structures display a high degree of linearity, making them suitable for use in multilevel digital as well as analog applications. If the levels of distortion are not low enough, a number of compensation techniques can be used to linearize either type of source. These include ciruit methods such as complementary distortion, negative feedback, selective harmonic compensation, and quasi-feedforward compensation, as well as the use of pulse position modulation (PPM) schemes.

1.3.12 Average Transmitted Power

The average power transmitted by the source affects both the transmitter and receiver in an optical fiber system. A large average power increases the temperature of the source, decreasing its reliability and hence, lifetime. Shot noise in the receiver is directly proportional to the signal level. A higher signal level will therefore increase the contribution of shot noise to the total receiver noise power. In order to achieve long repeaterless spans, however, a large amount of output optical power is needed. Thus, a suitable compromise must be made.

1.3.13 Receiver Sensitivity

Receiver sensitivity may be defined as the amount of optical power that must be present at the receiver input for it to correctly detect an impinging symbol. The type of line code employed in a communication system can have an effect on this parameter. Receiver noise increases with increasing preamplifier bandwidth. Since multilevel codes enable a decreased bandwidth to be used, a smaller amount of noise is coupled into the preamplifier and weaker signals can be detected. Multilevel codes can thus be said to increase the receiver sensitivity, enabling a larger repeater spacing and/or higher bit rate to be achieved. The work of Muoi and Hullet [11] and Petrovic [12] indicates that, at low bit rates, the use of a multilevel code requires considerably lower values of photodetector avalanche gain. This facilitates biasing and offers greater temperature stability, thus considerably simplifying the construction and implementation of APD detectors. On the other hand, binary block codes, increase the bit rate of the system and thus the preamplifier bandwidth required. The noise coupled into the receiver is larger, increasing the probability of an incorrect decision. These types of codes, therefore, decrease receiver sensitivity.

1.3.14 Dispersion Effects

Optical fibers cause pulses transmitted through them to be spread out in time, that is, to be dispersed. This can cause the tail of one pulse to continue into the time slot of the next, resulting in intersymbol interference. Due to their increased information capacity and resulting decrease in required transmission bandwidth, multilevel codes are not affected by dispersion to the extent that binary codes are. Thus, multilevel codes allow higher bit rates to be achieved before intersymbol interference starts to degrade performance.

1.3.15 Circuit Complexity

Economical and reliable implementation and maintenance of a communication system is enhanced when the coding and decoding circuits are fairly simple. In general, multilevel schemes will require more complex circuits than binary ones. However, if the proposed system is to be widely used, the required circuits can be integrated onto an IC package, thus obviating any concern with circuit complexity.

1.4 Types of Line Codes

Several coding schemes have been designed in an attempt to satisfy the criteria enumerated in the previous section. They fall into three basic categories:

1. codes which convert one bit into one coded symbol;
2. codes which convert several bits into one coded symbol;
3. codes which convert several bits into several coded symbols.

1.4.1 Simple Codes

Line codes with a one to one relationship between the coded and uncoded symbol are the simplest to implement. These codes may consist of two or three levels, and include, among others, the unipolar and bipolar formats.

The unipolar code transmits marks as pulses and spaces as absence of pulses, and is thus the least complex code. There are, however, significant disadvantages associated with it. No restriction exists on the number of consecutive identical symbols that can be transmitted and thus timing information may disappear. This problem can be overcome through the use of a scrambler, which randomizes the input signal by adding it modulo two (exclusive OR) to a bit stream from a pseudo random number generator of a fairly large sequence length. If the scrambling source has very low correlations at intervals of a word length or frame length (where the signal source may have strong correlations), the resulting signal is practically a stream of "independent random digits" [34]. Such a scrambling source can be implemented with a long shift register sequence which is relatively prime to both word and frame length. The use of such a scrambler ensures a large number of level transitions without the introduction of a power penalty. The power spectrum of a unipolar sequence contains significant energy at dc and low frequencies, causing the system to be subject to dc wander when ac coupling of the receiver amplifying stages is used. Also, error monitoring without interruption of service is impossible without a knowledge of the source statistics. A scrambled binary non-return-to-zero line code has been used in a commercial optical fiber system [35].

The alternate mark inversion (AMI) code is a three level (bipolar) code. Binary zeroes are coded as an absence of pulses, while binary ones are coded with alternating positive and negative pulses, the alternation taking place after each occurrence of a one. The power spectrum of an AMI encoded signal contains little dc and low frequency energy, permitting the use of ac coupling without any baseline wander. Errors can be detected by noting the occurrence of two consecutive ones with the same polarity. However, the timing problem of unipolar coding still exists, as no restriction is placed on the maximum length of a string of zeroes. Also, the efficiency of the code is greatly reduced (63 %) compared to that of straight binary (100 %), since the introduction of the third symbol does not increase the information capacity. In optical systems this code has been implemented as a two level code. Each of the three levels of conventional AMI is converted into two binary symbols, doubling the required transmission bandwidth [36].

1.4.2 Multilevel Codes

Codes that convert several binary symbols into one symbol are multilevel (n-level) pulse amplitude modulation (PAM) or pulse time modulation (PTM) schemes. Only PAM codes are considered below. For these codes, the information capacity of each symbol is increased and the required transmission bandwidth is decreased over the same properties in a binary code. Timing properties are slightly improved. Although an unrestricted sequence of zeroes can still be transmitted, it will be $\{1/\log_2 n\}$ the length of the corresponding binary sequence. In addition, framing is not required, and the code is 100 % efficient. Multilevel signals can increase the effective receiver sensitivity (compared to binary) and consequently can increase the attainable repeater spacing and/or bit rate [11,12] as shown in Figure 1.6. These curves show that two limits exist for bit rate and repeater spacing in an optical fiber transmission system. Maximum repeater spacing is determined by the attenuation in the fiber, if a low enough bit rate is used. If the bit rate exceeds a particular threshold (this threshold rate corresponds to the rate at the point where the horizontal and vertical curves join and is determined by the combination of fiber and line code used), the system becomes dispersion limited. Most optical systems are dispersion limited. Multimode fiber has a fairly high dispersion, while the single mode fiber used in long wavelength, high bit rate systems has such a low loss that the effect of dispersion is felt before that of attenuation. The use of a multilevel code raises the dispersion limit and allows a higher bit rate to be transmitted for a fixed repeater spacing, or the repeater spacing to increase for a fixed bit rate. The equations used to calculate these curves and the assumptions used in the calculations can be found in Appendix B. Multilevel codes have several disadvantages, however. Errors cannot be detected and the code is not bit sequence independent. If an incorrect symbol is received, then in the worst case, all $\{\log_2 n\}$ decoded binary bits will be wrong. The average transmitted power is greater than that of a binary signal and source nonlinearity is of concern in transmitter design. Experimental systems utilizing multilevel codes have been tested [12], but have not yet been used in commercial applications, to the author's knowledge.

1.4.3 Block Alphabetic Codes

The timing, spectral shape, and error monitoring problems encountered in the types of code described above can be reduced by the introduction of redundancy. Redundancy can be

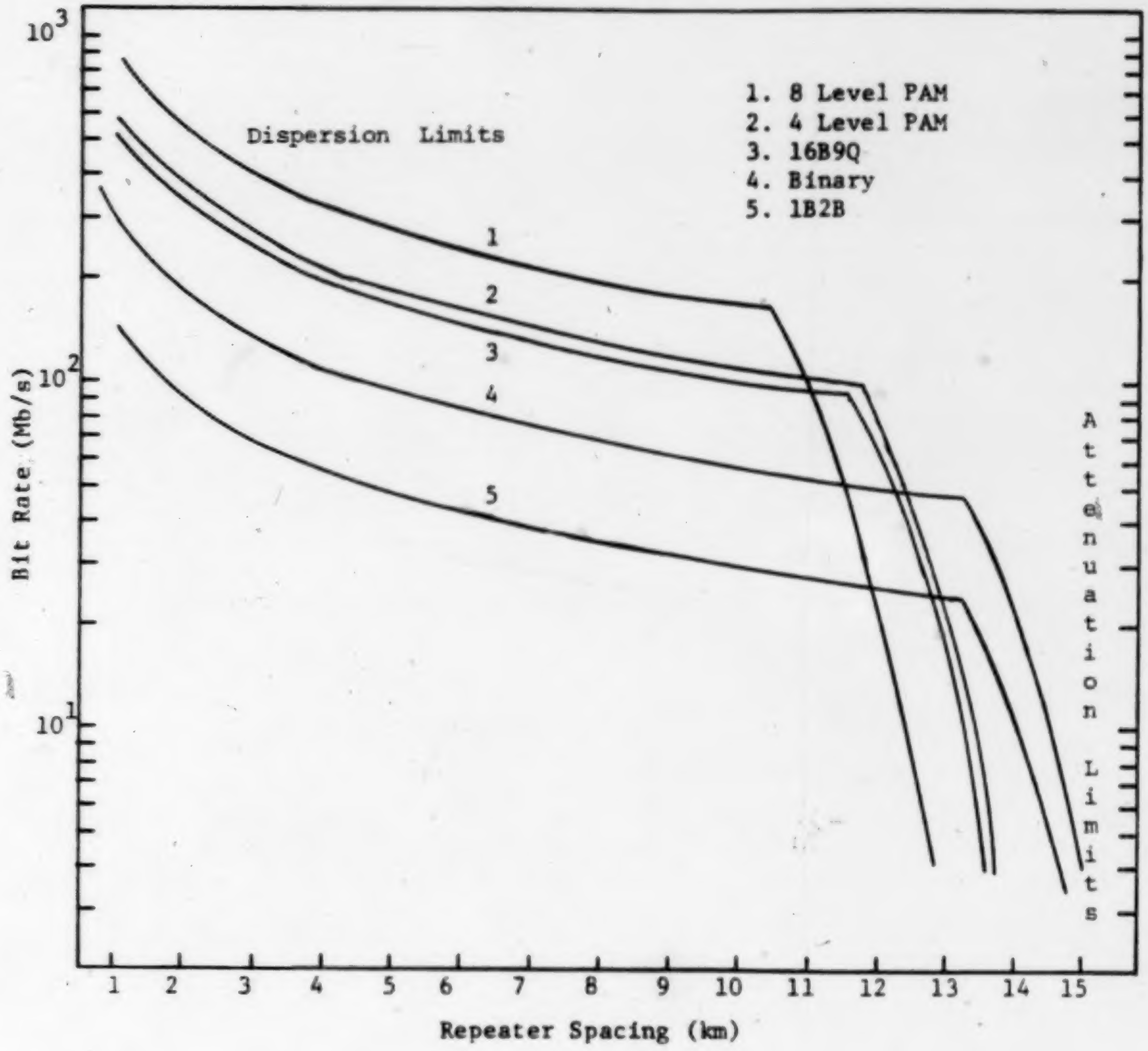


Figure 1.6. Theoretical bit rate versus repeater spacing curves for several line codes.

added to the coded signal by one of two methods: increasing the baud rate of the system (ie. sending more symbols) or increasing the number of pulse levels while keeping the baud rate constant. The third class of line code, block alphabetic codes, utilizes this feature.

These codes convert a block of binary information digits into a block of either binary or multilevel symbols. The mBnB codes (m binary bits to n binary bits, n>m) add redundancy to the signal by adding redundant bits. The bit rate is therefore increased, and the system requires a larger bandwidth. The mBnL codes (m binary bits to n L-level symbols, n<m) add redundancy by increasing the number of amplitude levels, allowing a decrease in the symbol rate and required bandwidth.

The efficiency of a code is defined by

$$\eta = \frac{\text{actual information rate}}{\text{theoretical maximum information rate}} = \frac{m \log_{10}(2)}{n \log_{10}(L)} \tag{1.1}$$

where m is the number of original binary digits and n is the number of L level symbols (this includes the mBnB codes, where L=2). A code becomes more efficient as 2^m approaches L^n ($2^m \leq L^n$ always). An efficient mBnB code is desirable in order to keep the increase in required bandwidth as small as possible, while an efficient mBnL code ($L > 2$) allows the information capacity to be increased to the greatest extent. However, the complexity of the codec (coder-decoder) increases greatly as redundancy is reduced [37]. The implementation of block codes is also more complex due to the necessity for frame recovery circuitry.

As the word "alphabetic" implies, these codes are designed as sets of alphabets or code word tables, with one or more code words corresponding to each input binary word. The redundancy of the code allows the alphabets to be designed so that they satisfy the requirements enumerated in section 1.3. Since $2^m < L^n$ generally, various assignments of code words to binary words are possible. There may be a number of unused code words (for example, 00...00 is never used), or several code words may be assigned to one binary word (the state of the system determines which code word is transmitted). Different code word assignments give rise to differences in power spectra, timing content, ease of frame recovery, and error detection.

Multilevel block codes (mBnL) are more advantageous than binary block codes (mBnB) for multimode optical fiber systems, since they reduce the bandwidth which the signal requires and increase the information capacity (while mBnB codes increase the bandwidth required for unchanged information rate). There is, as always in communication systems, a tradeoff. Multilevel codes have an additional power penalty as a result of the increased number of levels (and thus a degradation in signal to noise ratio), and multilevel transmitters are more complex. However, the effective receiver sensitivity increases for multilevel codes compared to binary ones (since the line signalling rate decreases) [11,12], so that the average impairment is small.

A major disadvantage of both mBnB and mBnL line codes lies in the complexity of the codec (coder-decoder). The assignment of coded blocks to uncoded binary blocks is arbitrary (that is, there is no general formula for this assignment); therefore, the encoding and decoding circuits must be implemented with Read Only Memories (ROM). As m and n increase, the size of the ROMs becomes prohibitively large and the complexity of the circuit needed to address individual locations increases greatly [38]. Table 1.2 shows the size of ROM needed for several values of n . The use of a ROM also limits the maximum speed of the circuit and thus the maximum bit rate.

Several mBnB codes have been implemented in optical fiber systems (for example, a 7B/8B code at 274 Mb/s by the Deutsche Bundespost)[39]. The mBnL codes have not been used in commercial optical fiber systems yet, to the author's knowledge. Figure 1.7 illustrates several line codes.

1.4.4 A New Code Classification

From the previous discussion, it can be seen that the implementation of a multilevel line code for optical fiber communications is advantageous. A code which combines the desirable characteristics of multilevel PAM and multilevel block (alphabetic) codes essentially overcomes the disadvantages of these two types of codes. Such a code has good timing and spectral properties, an increased information capacity, error detection capability, low error extension, and is transparent. The encoding and decoding processes are similar to those used in multilevel PAM, and thus the codec is fairly simple and fast (ROM look up is eliminated). The only disadvantages are the required increase in transmitted power and the consideration of source nonlinearity in transmitter design. These disadvantages are inherent in any multilevel PAM code, however and, depending on the application, should be a small price to pay for the

Table 1.2. ROM Size Needed for Various mBnB Line Codes.

n	ROM (bits/page)
1	8
2	64
3	384
4	2048
5	10240
6	49152
7	229376
8	1048576

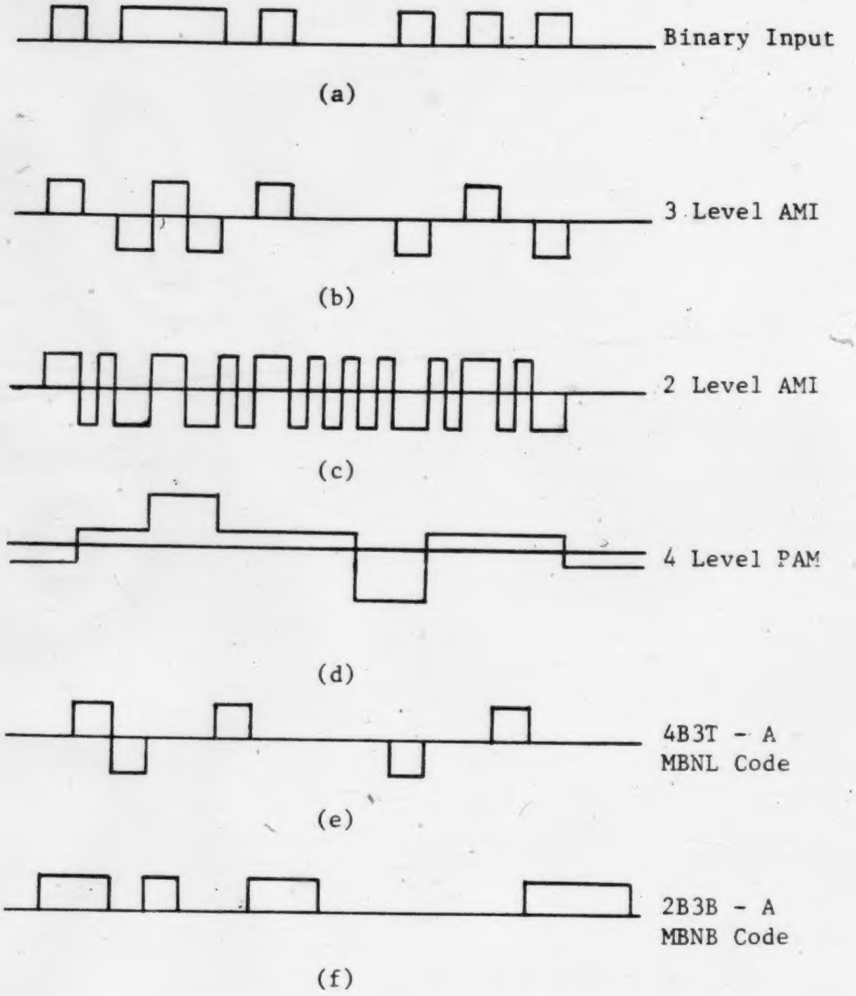


Figure 1.7. Examples of several line codes.

achievement of the remaining goals of line coding. A code with these properties can be classified as an unconventional multilevel block code. This thesis considers the use of such a line code in an optical fiber communication system.

1.5 Thesis Objectives and Organization

1.5.1 Objectives

Line codes employed in optical fiber communication systems to date have been binary and two level partial response codes. Multilevel codes have been characterized as impractical due to the nonlinearity of the optical sources as well as the more complex circuitry required for the transmitter and receiver. However, multilevel codes offer significant advantages over binary codes (increase of receiver sensitivity due to reduction of bandwidth requirement, ability to detect errors due to added redundancy, increased information capacity). It is therefore the purpose of this thesis to investigate a newly developed four level line code for implementation in an optical fiber communication system. The code studied is the 16B9Q code, which converts sixteen binary bits to nine quaternary symbols before transmission. The specific goals of the investigation are:

1. To design and implement a coder and decoder utilizing the 16B9Q line code for a multimode optical fiber system operating at 44.736 Mb/s. A complete end-to-end fiber link will be set up, and various power spectrum, timing content, and BER measurements will be made. This will show that it is practically feasible to use a fairly complex line code at a high bit rate.
2. To develop a statistical model of the line code.
3. To examine the power spectrum of the coded signal and to compare it to the theoretical power spectrum of the code, when generated numerically and analytically.
4. To deduce the timing content of the code from the experimental power spectrum and from theoretical considerations.
5. To show that the code is bit sequence independent.
6. To examine the effect of the variation of several system parameters on the error rate.

1.5.2 Organization

Chapter 1 has presented a general introduction to line coding, taking into account the specific constraints imposed by optical fiber systems. Chapter 2 will describe the 16B9Q line code in detail. In Chapter 3, a statistical model for the 16B9Q line code is developed, from which an analytical expression for the power spectrum is obtained. Chapter 4 describes the experimental system designed and constructed for this project, and includes a detailed description of the coding and decoding circuits. Chapter 5 discusses experimental results. The operation of the implemented coder and decoder is described, and it is verified that the use of the 16B9Q line code enables a greater transmission range to be obtained over dispersion limited fiber. The experimentally obtained power spectrum is compared to the analytically derived spectrum, as well as to one obtained by computer simulation. Bit sequence independence, timing content, and the effect of the variation of several system parameters upon the error rate are also discussed. Chapter 6, the concluding chapter, summarizes the results obtained and outlines problems and applications for future investigation.

CHAPTER 2

THE 16B9Q LINE CODE

2.1 An Unconventional Multilevel Block Code

In August 1983, engineers at Bell-Northern Research obtained a patent for the design and development of an unconventional multilevel block code for paired copper cable transmission systems [40]. This is a 2nB/(n+1)Q code, with the preferred realization being the 16B9Q code, in which sixteen binary bits are transformed into nine quaternary symbols. The characteristics of the code make it suitable for application to optical fiber communication systems.

Since the first commercial,optical fiber system was installed in 1976, the demand for optical communications has increased phenomenally. As a result, many fiber systems are now in place throughout the world. The majority of these systems operate at bit rates close to either 6.312 Mb/s or 45 Mb/s (DS-2 and DS-3 designation on the North American digital hierarchy). They use graded index, multimode fiber with a transmission bandwidth just large enough to transmit the operational bit rate, and all employ binary coding. As the demand for information capacity increases (due to normal growth or the request for new services) two alternatives exist to supply it. A new, higher bit rate system utilizing better fiber can be installed, or the existing system can be upgraded by changing the electronics at both ends while using the same fiber. The second alternative is the more economical one (since installing fiber is much more expensive than installing new electronics) [13] and can be implemented by changing the line code.

With a fixed fiber bandwidth, the use of multilevel signalling allows a higher bit rate to be transmitted. The greater the number of levels, the higher the bit rate for the same baud rate. As the number of levels increases, however, so does circuit complexity and consequently cost. A four level PAM code provides a 100% bit rate increase with a tolerable increase in circuit complexity. As noted earlier, though, four level PAM has no error detection capability and is not bit sequence independent. These two problems can be corrected by the addition of redundancy to the signal, but the bandwidth required for signal transmission will increase and the efficiency of the transmission will decrease.

The 16B9Q code provides a reasonable compromise. It compresses the transmission bandwidth to 56.25% of that required for plain binary signalling, and it is 88.9% efficient. Use of this code enables a 45 Mb/s signal to be transmitted over fiber which presently transmits 6.32 Mb/s (15 MHz bandwidth) or a 90 Mb/s signal to be transmitted over fiber currently carrying 45 Mb/s. The factor of three obtained in the first case is a result of the Nyquist theorem which states that a channel bandwidth of only half the signalling frequency is necessary in order to be able to recover all the data. Thus, half of (56.25% x 45 Mb/s) gives 12.66 Mb/s which can readily be transmitted over a fiber with a bandwidth of 15 MHz.

2.2 16B9Q - A 2nB/(n+1)Q Line Code

The 16B9Q four level code is a member of the family of codes designated as 2nB/(n+1)Q. These codes transform 2n binary bits into (n+1) four level symbols. They are multilevel block codes, but the transformation from binary to quaternary is not an arbitrary assignment as in the case of multilevel alphabetic block codes. Sequential rules are followed in order to convert the two level signal into a four level signal. The proper choice of n allows an optimal tradeoff of the code properties. With n=8, the 16B9Q code is obtained.

The most important code requirements can be translated into constraints placed upon the running digital sum (RDS) and the digital sum variation (DSV), which are defined as

$$RDS(k) = \sum_{i=1}^{k} C_n(i) \qquad C_n = -1,-3,1,3 \tag{2.1}$$

$$DSV(k) = RDS(k) - RDS(k-1) \tag{2.2}$$

where c_n represent the values of the quaternary digits, and k is the index (i.e. the kth word from the beginning of the transmission) of the coded word. The RDS is therefore the sum, from the beginning of the transmission to the present, of all the symbol values that have been transmitted. The DSV is the sum of the transmitted symbols over some finite period of time, usually over one coded word. Examples of RDS and DSV are shown in Figure 2.1.

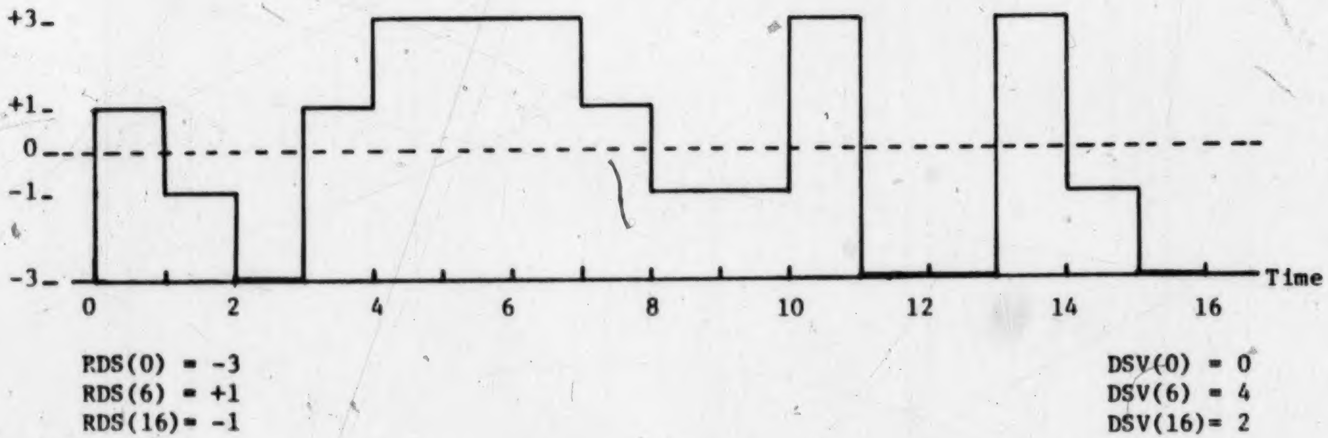


Figure 2.1. Examples of the RDS and DSV for a four level symbol stream.

Before proceeding further, the terms zero DC level, positive, and negative, need to be clarified with respect to optical signalling. In optical transmission, the term zero DC component signifies that the DC value of the signal remains at the mean light intensity. This value is just a reference value and does not contain any information about the signal. A non zero DC component indicates that the DC value of the signal has changed from the mean light level. The term "positive" refers to a pulse with an amplitude greater than the mean light intensity, while "negative" refers to a pulse with an amplitude lower than this intensity. The mean light intensity is the "0" level. The "0" level would be half of the maximum intensity. In an actual system, the four level digits would be represented by four unequally spaced light intensities (for optimum noise immunity and range) [11,12,41], or by four equally spaced light intensities (for circuit simplicity [41] or when using a PIN-FET receiver [8]) as shown in Figure 2.2. In the unequally spaced case, however, the c_n would have to be weighted differently (in order to keep the average RDS equal to zero), increasing the complexity of a practical code implementation.

By constraining the RDS to take on as few values as possible, and thus remain close to zero (ideally we want the RDS to equal zero), one constrains the DC and low frequency components of the signal. Timing information is also provided, since to keep the RDS low, many transitions from "positive" pulses to "negative" pulses must occur.

2.3 16B9Q Coding Rules

Several approaches exist through which 2n binary symbols can be converted into (n+1) quaternary symbols. Two of the simplest ones will be discussed below.

2.3.1 Monoblock Code

The first method deals with the monoblock code, in which 2n binary bits are converted to n quaternary symbols in one of two ways, and the (n+1)th quaternary symbol indicates the method of conversion. A four level symbol stream is created from a binary symbol stream by means of the following rules:

1. Convert 2n binary bits into n four level symbols by the same method used in normal four level PAM (i.e., see Table 2.1).

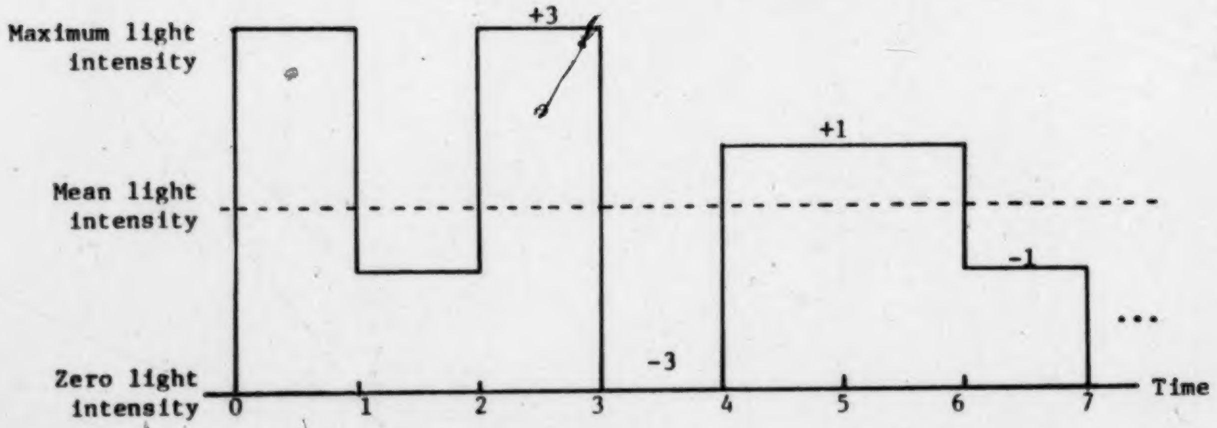


Figure 2.2. Relative light levels for a four level optical system.

Table 2.1. Translation of Binary to Quaternary Symbols.

Binary Input	Quaternary Output
01	-3
00	-1
10	+1
11	+3

2. Compute the DSV of the frame by summing the values assigned to each symbol in the four level frame.
3. Subtract 3 from the DSV. This adjusts the DSV in anticipation of the addition of the (n+1)th four level symbol to the frame. It also adds the correct value of the (n+1)th symbol to the final DSV and RDS, before that symbol is created.
4. If the polarity of the DSV is the same as the polarity of the RDS at the end of the previously transmitted frame, invert the word. That is, shift the pulse levels so that a +1 pulse becomes a -1, a +3 becomes a -3, and vice versa. If the polarities of the DSV and RDS are opposite, leave the word unchanged.
5. If the n symbol frame was inverted, add a positive pulse (+3) in the (n+1)th slot. If the frame was not inverted, add a negative pulse (-3) in the (n+1)th slot. The DSV does not have to be further adjusted, as the addition of -3 in step 3 and the subsequent inversion (if necessary) in step 4 have already caused the addition of the correct value of the ninth symbol to the DSV.

If the polarity of the code frames is chosen in this manner, the RDS will be constrained. An example of this process is shown in Figure 2.3.

The worst case RDS occurs with the appearance of a maximum excursion RDS frame (all +3 or all -3) after an initial RDS of ±3. This results in the transmission of a sequence of (n+1)(+3) or (n+1)(-3) symbols, with DSV and RDS magnitudes of $|3n+3|$ at the end of the frame. The following frame to be transmitted will have a DSV of opposite polarity (according to the rules listed above). However, the instantaneous RDS during the transmission of this frame can become greater than $|3n+3|$, since the initial pulses in the frame may increase the RDS before it decreases again. The maximum value of RDS occurs when the frame consists of n/2 pulses of the same magnitude and polarity as the RDS, followed by (n/2 + 1) pulses of the opposite polarity and magnitude for n even, or (n/2 +1/2) such pulses for n odd. This gives a maximum instantaneous RDS value of

$$3n + 3 + 3/2(n) = 4.5n + 3\ ,\ n \text{ even}$$

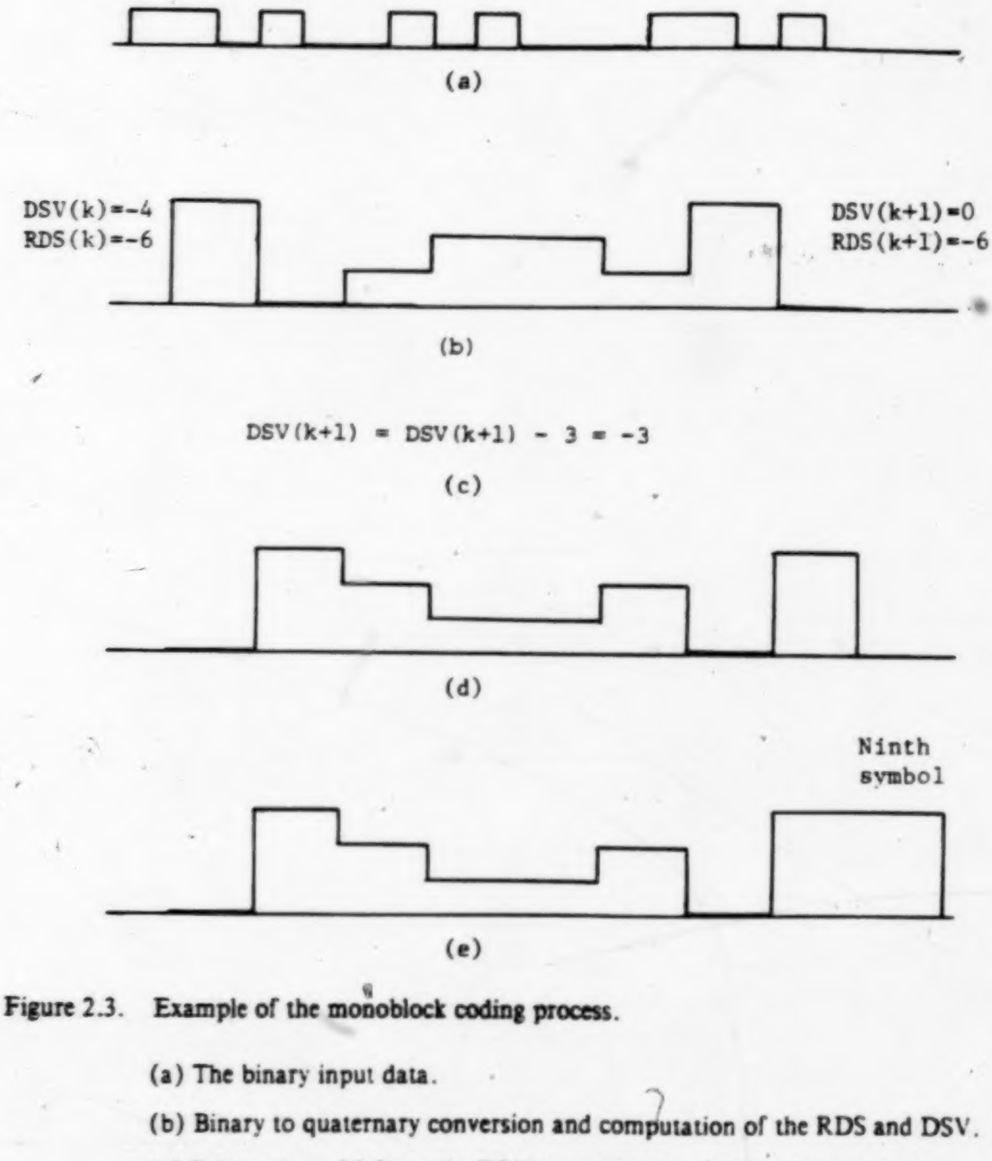


Figure 2.3. Example of the monoblock coding process.

(a) The binary input data.

(b) Binary to quaternary conversion and computation of the RDS and DSV.

(c) Subtraction of 3 from the DSV in anticipation of the ninth symbol.

(d) Inversion of the word (DSV[k + 1] and RDS[k] have the same polarity).

(e) Addition of the ninth symbol.

$$3n + 3 + 3/2(n) + 3/2 = 4.5n + 4.5 ,n \text{ odd}$$

Table 2.2 displays the maximum values of RDS at the end of a frame and the maximum instantaneous RDS values for various n. As n increases the maximum RDS increases, thus increasing the low frequency content of the signal. The value of n should be fairly small in order to constrain the low frequency content as much as possible. For small n, however, the efficiency of the code and the bandwidth reduction are low, as seen from Table 2.3. A reasonable compromise occurs for n=8.

2.3.2 Split Block Code

The second coding method, which results in the split-block code, utilizes the quaternary nature of the code to identify the coding options. Although a four level symbol can represent four states, the coding scheme described above only uses two of the four possible states to describe the ninth symbol (and thus the coding action performed). These two unused states can be used to decrease the maximum values of the RDS and DSV, thus decreasing the low frequency content of the transmitted signal.

If n is even, the n symbol four level frame is divided into two words of n/2 symbols. The DSV of the first word is calculated and 2 is subtracted in anticipation of the addition of its component in the ninth symbol. The polarity of the DSV is compared to the polarity of the RDS at the beginning of the word. If they are the same, the word is inverted before transmission; otherwise, it is transmitted unchanged. The DSV of the second word is calculated and 1 is subtracted from it, again in anticipation of the ninth symbol. The polarity of the DSV is compared to that of the RDS at the end of the first word. If the polarities are the same, the word is inverted. Finally, an (n+1)th symbol is added. It indicates whether neither, one or the other, or both of the two words were inverted. The amplitude and polarity of this pulse is such that the DSV of the pulse is the same as the numerical value of the numbers added to the DSVs of the words after the necessary inversions have occurred.

If the correspondence between the binary signals and the quaternary symbols is as shown in Table 2.1, then the (n+1)th symbol is calculated according to the rules shown in Table 2.4.

Table 2.2. Maximum RDS Values for Various Monoblock 2nB/(n+1)Q Line Codes.

n	RDS (end of frame)	RDS (instantaneous)
1	6	9
2	9	12
3	12	18
4	15	21
5	18	27
6	21	30
7	24	36
8	27	39
9	30	45
10	33	48

Table 2.3. Bandwidth Reduction and Efficiency for Various Monoblock 2nB/(n+1)Q Line Codes.

n	2nB/(n+1)Q	Bandwidth (%)	Efficiency (%)
1	2B2Q	100	50
2	4B3Q	75	66.7
3	6B4Q	66.7	75
4	8B5Q	62.5	80
5	10B6Q	60	83.4
6	12B7Q	58.3	85.8
7	14B8Q	57.1	87.4
8	16B9Q	56.3	88.8
9	18B10Q	55.6	90
10	20B11Q	55	90.9

Table 2.4. Rules for Calculation of the $(n+1)^{th}$ Symbol.

First Word	Second Word	$(n+1)^{th}$ Symbol
Invert	Invert	+3
Invert	Noninvert	+1
Noninvert	Invert	-1
Noninvert	Noninvert	-3

The advantage of this split block method is that, for any value of n, the RDS limits are much tighter than before, resulting in better control of the low frequency spectral content. The maximum value of the RDS at the end of a block is 3 + 3/2(n), while the maximum instantaneous value is 3 + 9/4(n). Table 2.5 compares the RDS values of the monoblock and split block coders for various values of n. For n=8 (16B9Q) the RDS bounds for the split-block coder are constrained to 54% of the bounds for the monoblock coder.

The code chosen to be implemented in this investigation, therefore, is the split block 16B9Q code.

2.4 Framing and Decoding

Decoding of the split block 16B9Q code relies on the recognition of the ninth symbol of the frame in order to determine whether inversion of any of the two, four symbol words has occurred. Once the ninth symbol has been found, the decoding procedure consists of inverting the appropriate words, removing the ninth symbol, and converting the quaternary symbols to binary using the inverse of the process used to code the original binary bits. Therefore, the coded signal must be framed before decoding can occur.

One possible framing procedure depends on finding the correct value of the RDS, thus requiring a knowledge of the allowable DSV values. The possible DSVs that each four symbol word can take on are 0, ±2, ±4, ±6, ±8, ±10, and ±12. Since each DSV must have a polarity, a DSV of 0 is assumed to be positive. This causes an ambiguity to exist in the coding rules since the polarity of the DSV does not change when a zero DSV word is inverted. With this exception, a positive DSV always follows a negative RDS and vice versa. The probability that this polarity alternation rule will be violated when the decoder is in an out of frame condition has been determined by computer analysis to be 0.3 [42]. When in frame, with an error rate of up to $1\text{x}10^{-3}$, the probability of violation is very low. Thus, the correct RDS value can be determined provided that the violation detection mechanism is suppressed when a zero DSV block is encountered. Correct framing is obtained when the correct RDS is found. Therefore, one method for finding frame is as follows:

1. Initialize the RDS counter to zero.
2. Assume a frame at random in the data stream and check for a polarity violation by

Table 2.5. Maximum RDS Values for the Monoblock and Split Block Versions of Several 2nB/(n+1)Q Line Codes.

	Monoblock		Split Block	
n	RDS End of Frame	RDS Instantaneous	RDS End of Frame	RDS Instantaneous
2	9	13	6	6
4	15	22	9	12
6	21	31	12	15
8	27	40	15	21
10	33	49	18	24

comparing the polarity of the DSV of the first word with the polarity of the RDS (which is set to 0 initially).

3. If a violation exists, modify the RDS in the following manner:

   RDS & DSV both positive : RDS = RDS - 1
   RDS & DSV both negative : RDS = RDS + 1

4. Update the RDS by adding to it the first word DSV. Then check the rear block for a polarity violation and modify the RDS as in (3).
5. Update the RDS with the second word DSV and the ninth symbol, and proceed to the next frame.
6. Continue to check the incoming frames, recording the number of violations in one counter, N, and the number of frames during which the violations occurred in another counter, M. Counter M sets counter N to zero after M frames.
7. If N violations occur in M frames, slip the frame by one symbol, and repeat the entire process for the new frame position.
8. When correct frame is obtained the RDS will quickly converge to its correct value and the violation count will fall to near zero (depending upon the error rate).

N and M are chosen, from statistical analysis of the code, in order to ensure a high probability of rapid reframe and a low probability of spurious reframe due to line errors. Since the occurrence of N violations in M frames determines an out of frame condition and the maximum RDS error at the end of a frame is 15, N must be greater than 15 in order to allow convergence of the RDS to its correct value. The probability of obtaining $>$ (N-15) errors in M frames must be $<$ .001 (for a maximum error rate of $1\text{x}10^{-3}$) in order to guarantee that correct frame is found in 99.9% of reframe attempts. If N=25 and M=256, the probability of finding frame is 99.98% for a $1\text{x}10^{-3}$ error rate [42].

2.5 Properties of the 16B9Q Code

2.5.1 Radix

The 16B9Q line code uses four amplitude levels. It offers significant advantages over binary codes in terms of information capacity and achievable bit rate, while keeping receiver and equalization circuit complexity low compared to that required for higher radix codes.

2.5.2 Spectral Shape

The 16B9Q code may be transmitted in a return-to-zero (RZ) or a non-return-to-zero (NRZ) format. These formats result in slightly different spectral energy distributions. The RZ code contains less total energy, but has a much higher level of high frequency energy than the NRZ code [42]. Nyquist's theorem states that all of the data can be recovered with a channel bandlimited to

$$f_n = \frac{f_s}{2} \tag{2.3}$$

where f_n is the Nyquist frequency and f_s is the signalling frequency. Thus, a code with little spectral content above f_n is desirable. Due to the constrained RDS, both the monoblock and split-block codes have zero energy at DC and at the symbol frequency. The monoblock code, however, has a larger amount of low frequency energy since the bounds on the RDS are wider. Both types of 16B9Q code contain a significant amount of energy at the Nyquist frequency, thus simplifying clock recovery (see Figure 1.6).

2.5.3 Timing Content

Timing extraction requires certain level transition density to be present in the received signal. The minimum number of transitions that can occur is one transition in 13 symbols for the monoblock code, and one transition in 8 symbols for the split block code. If the message sequence is not random enough, a scrambler can be used to increase this density.

2.5.4 Transparency

The 16B9Q code is essentially transparent once it has been properly framed. Since reframing depends on the randomness of the incoming data, a scrambler and descrambler may have to be used. The disadvantages of using a scrambler are an additional error extension factor of 3 [42] and increased circuit complexity.

2.5.5 Unique Decodability

Once the received signal sequence has been properly framed, the 16B9Q code is not only uniquely decodable but also instantly decodable. The ninth symbol in each code word specifies which portions of the word need to be inverted in order to obtain the original data stream. Thus, upon recognition of the ninth symbol, the input word is immediately known.

2.5.6 Redundancy

The 16B9Q code has a relatively high efficiency of 88.9%. It does not require complex circuitry and yet it provides good error monitoring capability and timing content.

2.5.7 Information Capacity

Due to the ninth symbol, which is added to the signal in order to permit decoding, each coded symbol represents 1.78 bits, compared to a maximum of 2 for a four level code.

2.5.8 Error Detection

In addition to signal coding (i.e. 16B9Q), the bits which constitute the signal stream also have to be coded. That is, the four combinations of two bits must be assigned in some way to each of the four signal levels. In order to minimize error extension, this is done in such a manner that only one bit changes between adjacent levels. One possible mapping is shown in Table 2.1. Assuming that a single error causes a change in state to an adjacent level only, a line error which occurs during the first eight symbols in a frame will result in a single bit in error. An error which occurs during the ninth symbol will cause one of three outcomes to occur. Either the first word, the second word, or the entire frame will be inverted by the decoder. Four bits will be wrong in the first two cases and eight in the third one, since the coding procedure inverts only half of the bits in a word when an inversion is called for. This same subset of bits is reinverted in the decoder, while the remaining bits pass through the system unchanged. A detailed explanation of the procedure is given in Chapter 4. Thus, the maximum error extension is 8, with an average error extension factor of 1.48 for equally likely errors in all levels [42]. The ratio of binary error rate to line error rate is .833, due to the ratio of symbol to bit rate. Thus, a line error rate of $1\text{x}10^{-3}$ is equivalent to a binary error rate (BER) of $8.33\text{x}10^{-4}$.

A line error in the four level code will result in an RDS error of 2 (due to the weighting of the four level symbols). The RDS will be corrected after two violations of the polarity rule are detected by the framing circuitry (in accordance with the reframe strategy described earlier), if a violation of opposite polarity does not occur first. Since the probability of error cancellation is less than 1% [42] for error rates up to $1\text{x}10^{-3}$, a fairly accurate error detector is obtained simply by dividing the output of the violation detector by two. The violation detector is the N counter described earlier.

Since the line error rate is much lower than the error rate when the decoder is out-of-frame, the value in the N counter does not reach its maximum in M frames. Therefore, spurious reframe is prevented by resetting this counter to zero every M frames.

2.5.9 Error Extension

As mentioned previously, the average error extension is 1.48. If a scrambler is used, it increases by a factor of 3.

2.5.10 Framing

Three parameters are important in the performance of block codes: time to recognize loss of frame, time to reframe, and mean time between spurious reframe.

The time to recognize loss of frame, T_L, is the mean time to detect N violations when out-of-frame.

$$T_L = \left(\frac{N}{2}\right) P_v \quad \text{frames} \tag{2.4}$$

where P_v is the probability of the occurence of a violation per block when out-of-frame.

The time for a 99% probability of reframe, T_R, is found from the probability of a slip occurring within K blocks (where a block is four symbols):

$$P_s(k \leq K) = \sum_{k=N}^{K} P_s(k) \tag{2.5}$$

where $P_s(k)$ is the probability of a slip occurring in exactly k blocks and is given by :

$$P_s(k) = P_v \frac{(k-1)!}{(N-1)!(k-N)!} P_v^{N-1} (1-P_v)^{k-N} \tag{2.6}$$

In the worst case the decoder could be eight symbols out-of-frame. Then

$$\{P_s(k \leq K\}^8 = 0.99 \tag{2.7}$$

or

$$P_s(k \leq K) = 0.99874 \tag{2.8}$$

Evaluation of this summation gives K and thus T_R. For $P_v = 0.3$ and $N = 25$, $K = 1197$ and $T_R = 0.19$ ms.

Spurious reframe occurs when the decoder is actually in frame but due to a large number of line errors, an out of frame condition is detected and the reframe procedure is initiated. The mean time between spurious reframe, T_S, depends on the line error statistics of the code and controls the possible values of the counter M for a specific value of N. The correspondence between binary symbols and quaternary symbols is designed such that a line error in the quaternary symbol will cause only one bit to be wrong in the two decoded binary bits. For $T_S > 1$ year, the probability of $> N/2$ errors in M frames must be less than 1/B, where B is the number of M frame periods in one year. That is,

$$P(\tfrac{N}{2}, M) \leq \left(\frac{M}{11.28} \right) \times 10^{-9}$$

For $N = 25$, $M = 256$, and a maximum error rate of 1×10^{-3}, $T_S > 1500$ years [42].

2.5.11 Source Nonlinearity

Current laser diodes and LEDs have fairly linear output characteristics. Therefore, the implementation of a four level code should not pose any problems.

2.5.12 Average Transmitted Power

If the probability of occurrence of each of the four levels is equal, the maximum level is set to be the same as the maximum level for the binary case, and all levels are equally spaced, the average transmitted optical power for the 16B9Q code will be the same as that for a binary code.

2.5.13 Receiver Sensitivity

Due to the four level signal and the added redundancy, the bandwidth necessary for a preamplifier for the 16B9Q line code is 56.25% of that required for a binary code. Thus, receiver noise is greatly decreased and the sensitivity of the receiver increases. The four level nature of the code, however, requires an increased signal to noise ratio over that for a binary code, in order to be able to accurately distinguish the levels. These two effects (reduced bandwidth and increased signal to noise ratio) tend to cancel each other. The net sensitivity of the receiver is therefore only slightly better than that for a binary code.

2.5.14 Dispersion Effects

Since the 16B9Q code is a four level code, the required transmission bandwidth is reduced. Therefore, a higher bit rate can be transmitted over a dispersion limited fiber before the signal is degraded to any significant extent.

2.5.15 Circuit Complexity

The implementation of the 16B9Q coder and decoder requires a large number of integrated circuits (ICs). However, the functions performed are not complex and lend themselves very well to semi-custom integration. The entire coder and decoder could each be integrated into one IC package, thereby greatly simplifying the practical implementation of the code.

CHAPTER 3

DERIVATION OF THE ANALYTICAL POWER SPECTRUM

One of the major functions of line coding is the modification of the power spectral density of a transmitted data sequence. This redistribution of the power in the frequency domain causes the signal to be more effectively matched to the transmission channel characteristics. A large number of useful line codes have been developed, and new ones will, no doubt be discovered in the future. When designing a transmission system, a choice must be made from amongst these codes, each of which has desirable characteristics. A knowledge of the power spectral density of each code permits their spectra to be compared and facilitates the selection of the most appropriate one for a given application. It is therefore important to find an expression for the power spectral density of a line code.

Following the initial effort to develop a method for spectrum evaluation of line encoded messages by Bennett in 1958 [45], little further work appeared in the literature for over a decade. Several authors [46-50] then considered various methods, but it was Cariolaro and Tronca [51,52] in 1974, who formulated the problem in the algebraic setting of automata theory and obtained a closed form expression. The algorithm developed by Cariolaro and Tronca, and utilized by Lindholm [53] and Poo [54] among others, is applied here to the 16B9Q line code to obtain an expression for its theoretical power spectral density.

3.1 Markov Processes

The line coding process subjects input binary digital data to a nonlinear transformation which can be modelled as a finite state sequential machine (FSSM) [50-55], also known as a Mealy machine. Before the power spectrum can be evaluated, the behaviour of the coded process must be known. This is determined from a knowledge of both the input signal characteristics and the coding transformation. Since the source data are random, a statistical analysis must be performed on the line coded signal. The initial problem, therfore, is to determine the statistics of the coded sequence, given the statistics of the source sequence.

The input sequence to the coder is a stochastic process, and is assumed to be stationary and memoryless. That is, it consists of random, mutually independent and identically distributed symbols. As mentioned above, the coder is a Mealy machine. Its output, therefore, depends on the input and on the state of the machine, and is classified as a Markov process.

Specifically, if the dependence between a succession of random variables is such that the future of the process depends only on the present state and not on past history, the process is said to be a Markov chain. This can be stated more formally as follows:

> A stochastic process $\{X(t),\ t\epsilon T\}$, where $X(t)$ is a random sequence and T is the set of observation times, is a Markov process if for any set of $n+1$ values $t_1<t_2<,\ldots,t_n<t_{n+1}\epsilon T$ and any set of $n+1$ states $\{x_1,\ldots,x_{n+1}\}$ [10],

$$P[X(t_{n+1}) = x_{n+1} | X(t_1) = x_1,\ X(t_2) = x_2, \ldots,\ X(t_n) = x_n]$$

$$= P[X(t_{n+1}) = x_{n+1} | X(t_n) = x_n] \qquad (3.1)$$

The conditional probabilities, $p_{ij}(n)=P[X(t_{n+1})=j|\ X(t_n)=i]$ are called state transition probabilities, and in general may depend on n. The next state probabilities $p_j(n+1)$ are the probabilities that the system will occupy state j at time n+1 given that it is in state i at time n, and are equal to the transition probabilities. The conditioning on these probabilities can be removed by averaging over all states at time n to obtain

$$p_j(n+1) = \sum_i p_i(n)\ P[X_{n+1} = j | X_n = i] \qquad (3.2)$$

where $p_j(n)$ is the present state probability.

Equation (3.2) is valid for all j. It therefore represents a set of equations which may be expressed in matrix form as

$$p(n+1) = p(n)\ Q(n) \qquad (3.3)$$

In an N-dimensional state space, $\mathbf{p}(n)$ and $\mathbf{p}(n+1)$ are N-dimensional row vectors and $\mathbf{Q}(n)$ is the NxN transition probability matrix. If the transition probabilities, $p_{ij}(n)$ are independent of time, i.e., if they depend only on the size of the step taken (in this case it is one - from n to n+1) and not on the time when that step occurred, then the transition probabilities, as well as the Markov chain, are stationary or time-homogeneous. Equation (3.3) can then be written as

$$p(n+1) = p(n)Q \qquad (3.4)$$

The elements of the transition probability matrix **Q** are probabilities, and its rows are conditional probability distributions (row $i+1$ gives the probability distribution of X_{n+1} if $X_n = i$). Therefore, all the elements of **Q** must be positive and the elements in each row must sum to unity (since some transition must always occur). Thus,

$$\begin{aligned} & p_{ij}(n) \geq 0 \quad \text{all } i,j \quad n = 0,1,\ldots \\ & \sum_j p_{ij}(n) = 1 \quad \text{all } i \quad n = 0,1,\ldots \end{aligned} \tag{3.5}$$

are always satisfied. Such a matrix is a stochastic matrix [56] and any stochastic matrix can be regarded as the matrix of transition probabilities of some homogeneous Markov chain. Given Q and an initial state probability vector **p**(0), the probability vector **p**(n) can be determined for any n from

$$p(n) = p(0)\, Q^n \tag{3.6}$$

For certain Markov chains, termed irreducible chains, there exists an equilibrium probability distribution such that, given any valid initial distribution, the system will always converge to equilibrium after a finite number of transitions. The limiting distribution is denoted as $\overset{\infty}{\mathbf{p}}$ and can be obtained from any valid initial state in a single step through

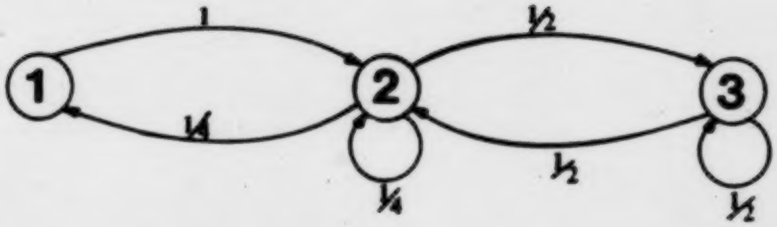
$$\overset{\infty}{p} = p(0)Q^{\infty} \tag{3.7}$$

where

$$Q^{\infty} = \lim Q^n \tag{3.8}$$

A Markov chain is irreducible if all states i can go to all states j, and vice versa in a finite number of steps.

A Markov chain can be easily visualized from a state transition diagram, as shown in Figure 3.1. This graph of interconnected vertices is a compact and useful representation of the process. The vertices correspond to states, while the directed edges denote allowable transitions and are thus labelled with the appropriate transition probabilities. Figure 3.1 illustrates a three



$$Q = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Figure 3.1. Example of a three state Markov process.

(a) The state transition diagram.

(b) The associated state transition probability matrix.

state Markov chain by means of its state transition diagram and transition probability matrix. It shows that state 1 is always followed by state 2, which can then go to states 1 or 3 with probability 1/4 or remain in state 2 with probability 1/2. Once in state 3, the next transition can be to state 2 (probability 1/2), or the system can again remain in its current state, 3 (probability 1/2). All states can be reached from any initial state in a finite number of transitions; therefore, this is an irreducible Markov chain. The transition probability matrix is stochastic, since all the elements are positive, and the elements in each row sum to one.

3.2 Power Spectrum of a Stochastic Process

The input data to a line coder is a stationary random process with equiprobable independent binary digits. A non-redundant line code will transform these data into a stationary random line signal composed of independent digits having simple statistical properties. This does not happen for block codes, however. Given independent, equiprobable source blocks, a block code produces equiprobable, independent line blocks. The symbols forming each block are not equiprobable or independent, in general, and the probability distribution of each symbol position in the block and the joint distribution of symbol pairs is determined by the code structure. In block codes with more than one alphabet (an alphabet is a code word look-up table) not even the line blocks are independent since the translation of any source block depends on past history. The statistics of the line signal are cyclostationary rather than stationary, since they vary periodically with a period equal to the length of a block, rather than being independent of time origin. Due to this cyclostationarity, care must be taken in deriving a valid spectral characterization.

3.2.1 Some Definitions

3.2.1.1 Stationarity

A process is strict sense stationary if all of its statistical properties are invariant to a shift of the time origin, i.e. $X(t)$ and $X(t+c)$ have the same statistics for any c. A less demanding form of stationarity requires only that the mean and autocorrelation of the process be time invariant:

$$E\{(t)\} = \overline{X} = \text{constant}$$

$$(3.9)$$

$$E\{X(t)\ X(t+\tau)\} = R_X(\tau) = R_X(\tau) = \text{function only of the time difference}$$

where E{ } denotes expectation or ensemble average, X is the time averaged mean, and $R_X(\tau)$ is the autocorrelation function. Such a process is called wide sense stationary. The autocorrelation function of a wide sense stationary process has the further properties that

$$R_X(\tau) = R_X(-\tau)$$

$$(3.10)$$

$$|R_X(\tau)| \leq R_X(0) = \overline{X^2}$$

3.2.1.2 Ergodicity

An ergodic process is a stationary process with the additional property that the ensemble average (E{ }) is equal to the time average, where the time average is defined as

$$A\{\xi_r\} = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{r=-N}^{N} \xi_r \qquad (3.11)$$

for a stochastic sequence.

3.2.2 Power Spectrum

For a wide sense stationary process, the Fourier transform of the autocorrelation function is the power spectrum or power spectral density,

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau)\ e^{-j2\pi f t}\ d\tau \qquad (3.12)$$

where $S_X(f)$ is real, even, and non-negative. For a particular realization of the process, this expression can be derived as follows [10]. Consider a truncated function $x_T(t)$

$$x_T(t) = \begin{cases} x(t), & |t| < T/2 \\ 0 \;, & \text{elsewhere} \end{cases} \tag{3.13}$$

such that

$$x(t) = \lim_{T\to\infty} x_T(t) \tag{3.14}$$

The truncation allows the Fourier transform to be taken

$$x_T(f) = \int_{-\infty}^{\infty} x_T(t)e^{-j2\pi ft}\,dt \;\leftrightarrow\; x_T(t) = \int_{-\infty}^{\infty} x_T(f)e^{j2\pi ft}\,df \tag{3.15}$$

and the energy to be written as

$$E_T = \int_{-\infty}^{\infty} |x_T(t)|^2 dt = \int_{-T/2}^{T/2} |x(t)|^2 dt$$

$$= \int_{-\infty}^{\infty} |x_T(f)|^2 dt \text{ by Parseval's theorem} \tag{3.16}$$

The power in x(t) in the time interval (-T/2<t<T/2) is

$$P_T = \frac{E_T}{T} = \int_{-\infty}^{\infty} \frac{1}{T} |x_T(f)|^2 df \tag{3.17}$$

The function under the integral describes the distribution of power with frequency; therefore, the power spectrum for x(t) is

$$S_x(f) = \lim_{T\to\infty} \frac{1}{T} |x_T(f)|^2 \tag{3.18}$$

Equation (3.18) applies to a particular realization of the process, and therefore it is a random variable. By first averaging over all realizations and then performing the limiting operation, a characterization for the entire process is obtained:

$$S_x(f) = \lim_{T\to\infty} \frac{1}{T} E\{|x_T(f)|^2\} \tag{3.19}$$

Equation (3.19) can be written in terms of autocorrelation functions as follows:

$$S_x(f) = \lim_{T\to\infty} E\{\frac{1}{T} \int_{-T/2}^{T/2} x(t_1)e^{-j2\pi f t_1}\, dt_1 \int_{-T/2}^{T/2} x(t_2)e^{-j2\pi f t_2}\, dt_2\}$$

$$= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \int_{-T/2}^{T/2} E\{x(t_1)x(t_2)\}e^{-j2\pi f(t_2-t_1)} dt_2 dt_1$$

$$= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \int_{-T/2}^{T/2} R_x(t_1,t_2)e^{-j2\pi f(t_2-t_1)} dt_2 dt_1$$

$$= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \int_{-T/2}^{T/2} R_x(t,t+\tau)dt\; e^{-j2\pi f\tau}\, d\tau$$

$$= \lim_{T\to\infty} \frac{1}{T} \cdot \int_{-T/2}^{T/2} A\{R_x(t,t+\tau)\}\; e^{-j2\pi f\tau}\, d\tau \tag{3.20}$$

It can be seen that the power spectrum and the time averaged autocorrelation function comprise a Fourier Transform pair

$$S_x(f) \leftrightarrow A\{R_x(t,t+\tau)\} \tag{3.21}$$

This relationship holds for any process, stationary or non stationary, since nothing was assumed about the time dependence of the autocorrelation in the derivation.

If $X(t)$ is wide sense stationary, then $A\{R_x(t,t+\tau)\} = R_x(t,t+\tau) = R_x(\tau)$ for all t and we have

$$S_x(f) \leftrightarrow R_x(\tau) \tag{3.22}$$

which is the same as Equation (3.12) and is known as the Wiener-Khintchine theorem. Equation (3.21) therefore, can be used to obtain the power spectrum of a non stationary process. Although the time averaging operation causes some information to be lost, this average power spectrum is nevertheless sufficient for the description of line codes, and it has the same properties as the power spectrum of a stationary process. In the particular case of interest, cyclostationarity, $R_X(t,t+\tau)$ is periodic in time over an interval equal to the length of the coded word, and therefore it is sufficient to average over just one of these periods. The limiting process of Equation (3.20) is then no longer necessary and we then have

$$A\{R_x(t,t+\tau)\} = \hat{R}_x(\tau) \triangleq \frac{1}{T}\int_{-T/2}^{T/2} R_x(t,t+\tau)dt \tag{3.23}$$

Thus, by using Equations (3.23) and (3.21), a proper power spectral density for a cyclostationary process is obtained.

3.3 Finite State Sequential Machine Model of the 16B9Q Code

We can now statistically model the 16B9Q line code in terms of the preceeding theory and derive its power spectral density.

Although the implementation of the 16B9Q line code does not involve the use of a code word look-up table, it is nevertheless a block code and can be modelled as an alphabetic block code (i.e., one that involves code word look-up). As shown in Table 3.1, the input words to the coder are sixteen bits long, so there are 2^{16} = 65,536 different combinations (a complete listing of the input words would require 1093 pages, assuming 60 lines per page). Four output alphabets, consisting of 65,536 four level nine symbol words exist. The four alphabets correspond to the cases when the first half, the second half, neither, or both halves of the input word are inverted. Thus, 16 input bits are converted into nine output symbols, the resulting code word sequence being a 9-dimensional discrete valued stochastic process, the statistics of which depend on the input symbol sequence statistics and on the encoder rules.

The first step in the derivation is the translation of the coder rules into a FSSM model. In general, a FSSM can be fully described by three sets and two functions:

1. The input set B of M digit binary input words. The number of elements in B is $|B| = 2^M$

Table 3.1. Code Word Translation Table for the 16B9Q Line Code.

Input	Alphabet 1	Alphabet 2	Alphabet 3	Alphabet 4
0000000000000000	+1+1+1+1+1+1+1+1+3	+1+1+1+1-1-1-1-1+1	-1-1-1-1+1+1+1+1-1	-1-1-1-1-1-1-1-1-3
0000000000000001	+1+1+1+1+1+1+1-1+3	+1+1+1+1-1-1-1+1+1	-1-1-1-1+1+1+1-1-1	-1-1-1-1-1-1-1+1-3
0000000000000010	+1+1+1+1+1+1+1+3+3	+1+1+1+1-1-1-1-3+1	-1-1-1-1+1+1+1+3-1	-1-1-1-1-1-1-1-3-3
0000000000000011	+1+1+1+1+1+1+1-3+3	+1+1+1+1-1-1-1+3+1	-1-1-1-1+1+1+1-3-1	-1-1-1-1-1-1-1+3-3
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1111111111111110	-3-3-3-3-3-3-3+3+3	-3-3-3-3+3+3+3-3+1	+3+3+3+3=3-3-3+3-1	+3+3+3+3+3+3+3-3-3
1111111111111111	-3-3-3-3-3-3-3-3+3	-3-3-3-3+3+3+3+3+1	+3+3+3+3-3-3-3-3-1	+3+3+3+3+3+3+3+3-3

= K. Thus, B is a KxM matrix composed of K, M-dimensional row vectors. A particular input word is denoted by b_n (i.e. the n^{th} word in the input sequence).

For the 16B9Q code, the input set consists of 65,536 different 16 digit binary words.

2. The set C of coder states. C is a finite set containing J elements. The state at the time the nth code word is translated is referred to as c_n .

   The set of states for the 16B9Q code is defined by the values that the RDS can take on. In Chapter 2 it was mentioned that the minimum and maximum RDS values are -15 and +15. However, due to the method in which the coder was realized, these bounds are even tighter. The initializations of the DSV adder in the coder to a -2 at the beginning of the first word and a -1 at the beginning of the second word in the frame, constrain the extreme values of RDS to -13 and +12. Since the RDS can take on any integer value between these two limits, there are thus 26 states.

3. The output set A of N digit L level (L-ary) code words. Each input word is translated into an output word , depending on the state of the encoder. Thus, the size of A is JK. In general, not all of these words are distinct since the same word may be used in more than one state. Similar to the matrix of input words, the code words are regarded as N-dimensional row vectors, and they are arranged in K matrices, A_u , of dimensions JxN, one matrix for each input word. These matrices are in turn arranged in the matrix

$$A = \begin{vmatrix} A_1 \\ \vdots \\ A_K \end{vmatrix} , \quad K = 2^M \tag{3.24}$$

   The 16B9Q output set is composed of 9 digit 4 level code words. Each input word can enter the coder when it is in any of the 26 possible states. For each state of each input word there is an output word. Therefore, the output set has 65,536x26 elements, but only 65,536x4 are unique.

4. The output function h which specifies the translation

$$a_n = h(c_n, b_n). \tag{3.25}$$

That is, given an input word and a state, an output word is produced. Cattermole [10] states that h must be chosen in such a manner that it allows state independent decoding. That is, given a code word a_n, one must be able to identify b_n without knowledge of c_n, in order to ensure unique decodability. Unique decodability is essential, but it does not imply or require state independent decodability. The practical implementation of a code which requires knowledge of the state as well as the code word for the decoding process will be more complex than that of a state independent decodable code, but it is not impossible. Therefore, this restriction upon h is not strictly necessary.

The 16B9Q output function is the rule that chooses which of the four unique output words will be transmitted for the particular input word. It is based on the initial RDS, and the DSV of the input word, and is in reality the coding rules outlined in Chapter 2. In Chapter 2 each input word was divided into two, and each half was then coded based upon the initial RDS before the half word and the DSV of the half word. Finally, a ninth 4 level symbol was added to indicate what had been done to the two previous halves. Here, these three operations are condensed into one. Since the RDS at the beginning of an input word can be any of the 26 possible values, the coding process can result in any of the four possible outcomes (first half, second half, neither, or both inverted). Thus, for each input word, there are four distinct alphabets from which the output word can come, as shown in Table 3.1. The process of condensing the three operations which actually occur in the coder into one, allows us to model the 16B9Q code as a (16,9) block code (16 bits are transformed into 9 symbols).

5. The state transition function g which specifies the state the encoder is left in after translating one word, i.e.

$$c_{n+1} = g(c_n, b_n) \tag{3.26}$$

The state transition function can also be represented in matrix form. For each input word b_u $(u=1,\ldots,K)$, a square matrix E_u of dimension JxJ is defined. The (i,j) entry of E_u is equal to one only if b_u causes the system to go from state i to state j; otherwise it is zero. For every present state and input word, the next state is completely known. Therefore, each row of E_u contains only one non zero entry, and that entry is a one.

The state transition function of the 16B9Q line code is the DSV of the 9 symbol code word. This DSV is added to the initial RDS to determine the final RDS, which is the next state.

The functions g and h are the characterizing functions of the FSSM and can be specified by a transition table in which the intersection of row b_i and column c_j contains the corresponding code word and next state. A more suggestive representation of these functions is given by the state transition diagram. Figure 3.1 is an example of such a diagram for a system with three states. The 16B9Q code has 26 states, therefore its transition diagram has 26 vertices, and 26 lines enter and leave each vertex. The diagram is thus very complex and in fact difficult to read. In this case therefore, the state transition matrix is more useful. The sequence of states that the coder passes through is a discrete random process due to the fact that the input words are independent and stationary. This sequence of states is also a Markov chain since the next state at any time depends only on the current state and the input. Anything that happened to reach the current state has no effect whatsoever on the probability distribution of the next state. Thus we have a matrix Q of transition probabilities p_{ij}

$$p_{ij} = P[X_{m+1} = j \mid X_m = i] \tag{3.27}$$

and this matrix converges to a limiting state distribution $\bar{\bar{p}}$.

3.3.1 State Probabilities

To find the state transition probabilities p_{ij}, all the code words which cause state i to change to state j must be identified, their occurrences summed, and this total divided by the number of total possible input words. In other words

$$p_{ij} = \frac{\sum b_{ij}}{\text{Total number of input words}} = \frac{|B_{ij}|}{|B|} \tag{3.28}$$

where b_{ij} is a word which causes a transition from state i to state j. If this is done for each initial state i $(i=1,\ldots,J)$ then the matrix Q is obtained. This is simply a process of counting words, and is very easy to do for a code with few input words. The 16B9Q line code, however, has 65,536 possible input words; therefore, a computer program was written to determine the transition probabilities. The program listing can be found in Appendix C. A short description of the procedure is given here.

In theory, the coding process would have to be carried out on every possible input word for every initial RDS to find the final RDS, and the number of times each final RDS occurred would be counted. This would involve 26x65,536 operations. The actual method used in this thesis is considerably simpler. First of all, the actual coding process is carried out on 8 bits at a time. Therefore, there are only $2^8 = 256$ possible input combinations. Secondly, there are only 13 discrete DSV values that each 8 bit word can assume. Without initialization, these values are the even integers between -12 and +12 inclusive. Since the first word and the second word in an input frame are initialized to -2 and -1 respectively, the actual DSVs are altered by this constant amount. A computer program (Program 1 in Appendix C) was written to generate the 256 input possibilities, calculate their DSVs, and count the number of times each DSV occurred. Since each input word is independent and equiprobable, this number divided by 256 (the total number of possible words) gives the probablility of the occurrence of the particular DSV. Table 3.2 shows the first and second word DSVs along with their probability of occurrence, while Figure 3.2 is a plot of the probability distribution function (pdf) of the DSV.

A second computer program (Program 2 in Appendix C) uses the pdf of the DSV as input to find the transition probabilities p_{ij}. For every initial RDS value, and for every possible first word DSV, the coding process is carried out for every possible second word DSV, as shown in Figure 3.3. For each first word and second word DSV combination, the probabilities of the two DSVs are multiplied together and added to the sum for the final DSV obtained. T[illegible] sum is p_{ij} - the probability that starting with initial RDS = i, and given a random input, the coding process will cause the final RDS to be j after the 16 bit word has been coded. Although the total possible number of 16 bit input words for each initial RDS is 65,536, only 169 operations are required since there are only 13 distinct values for each of the first word and second word DSVs. The multiplication by the probability of occurrence of each DSV takes care of the remaining 65,367 possibilities. Thus, a 26x26 transition probability matrix, Q, is obtained as shown in Figure 3.4.

The limiting distribution, $\overset{\infty}{\overline{p}}$, which is a column vector, each element, $\overset{\infty}{\overline{p}}_i$, of which gives the absolute probability of being in state i, can be obtained by solving the system of linear equations

Table 3.2. DSV Probabilities

DSV	Initialized First Word DSV	Initialized Second Word DSV	256xp_{DSV}
12	10	10	1
10	8	9	4
8	6	7	10
6	4	5	20
4	2	3	31
2	0	1	40
0	-2	-1	44
-2	-4	-3	40
-4	-6	-5	31
-6	-8	-7	20
-8	-10	-9	10
-10	-12	-11	4
-12	-14	-13	1

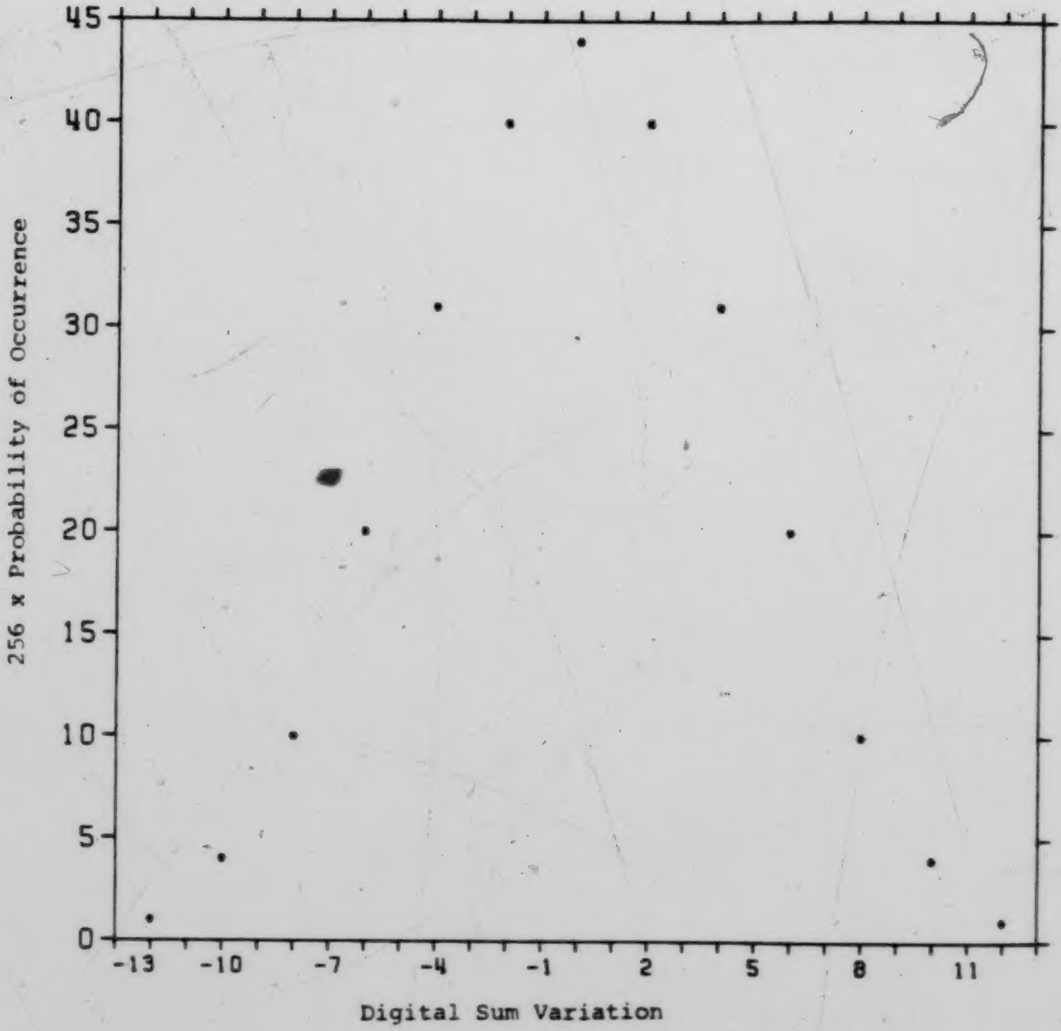


Figure 3.2. Probability distribution function of the 16B9Q line code DSV.

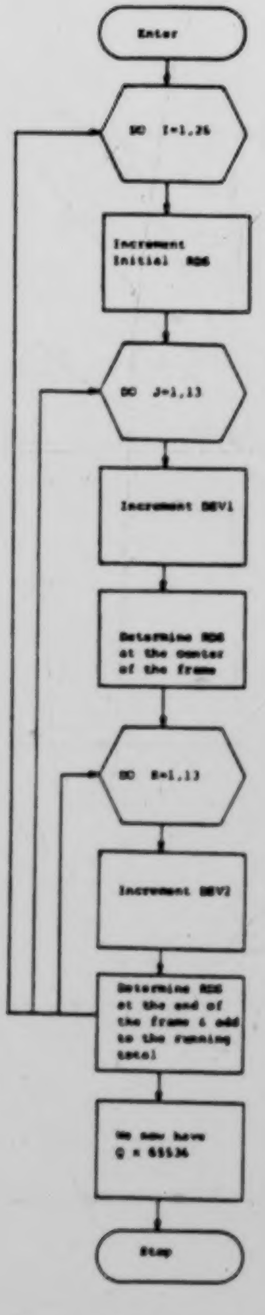


Figure 3.3. Flowchart of the process used to determine the transition probability matrix of the 16B9Q line code.

$$\begin{bmatrix}
0 & 3361 & 0 & 9145 & 0 & 12419 & 0 & 12759 & 0 & 10848 & 0 & 7840 & 0 & 4910 & 0 & 2514 & 0 & 1135 & 0 & 435 & 0 & 135 & 0 & 31 & 0 & 4 \\
4 & 0 & 3381 & 0 & 9201 & 0 & 12539 & 0 & 12963 & 0 & 11132 & 0 & 8176 & 0 & 4574 & 0 & 2230 & 0 & 931 & 0 & 315 & 0 & 79 & 0 & 11 & 0 \\
0 & 4 & 0 & 3381 & 0 & 9201 & 0 & 12539 & 0 & 12963 & 0 & 11132 & 0 & 8176 & 0 & 4574 & 0 & 2230 & 0 & 931 & 0 & 315 & 0 & 79 & 0 & 11 \\
11 & 0 & 59 & 0 & 3535 & 0 & 9531 & 0 & 13100 & 0 & 13744 & 0 & 12056 & 0 & 7252 & 0 & 3793 & 0 & 1669 & 0 & 601 & 0 & 161 & 0 & 24 & 0 \\
0 & 11 & 0 & 59 & 0 & 3535 & 0 & 9531 & 0 & 13100 & 0 & 13744 & 0 & 12056 & 0 & 7252 & 0 & 3793 & 0 & 1669 & 0 & 601 & 0 & 161 & 0 & 24 \\
24 & 0 & 131 & 0 & 395 & 0 & 4255 & 0 & 10755 & 0 & 14804 & 0 & 15760 & 0 & 10040 & 0 & 5548 & 0 & 2569 & 0 & 949 & 0 & 265 & 0 & 41 & 0 \\
0 & 24 & 0 & 131 & 0 & 395 & 0 & 4255 & 0 & 10755 & 0 & 14804 & 0 & 15760 & 0 & 10040 & 0 & 5548 & 0 & 2569 & 0 & 949 & 0 & 265 & 0 & 41 \\
41 & 0 & 229 & 0 & 705 & 0 & 1625 & 0 & 6346 & 0 & 13666 & 0 & 18248 & 0 & 12316 & 0 & 7129 & 0 & 3457 & 0 & 1339 & 0 & 375 & 0 & 60 & 0 \\
0 & 41 & 0 & 229 & 0 & 705 & 0 & 1625 & 0 & 6346 & 0 & 13666 & 0 & 18248 & 0 & 12316 & 0 & 7129 & 0 & 3457 & 0 & 1339 & 0 & 375 & 0 & 60 \\
60 & 0 & 341 & 0 & 1089 & 0 & 2505 & 0 & 4685 & 0 & 10606 & 0 & 18706 & 0 & 13208 & 0 & 8056 & 0 & 4069 & 0 & 1657 & 0 & 499 & 0 & 75 & 0 \\
0 & 60 & 0 & 341 & 0 & 1089 & 0 & 2505 & 0 & 4685 & 0 & 10606 & 0 & 18706 & 0 & 13208 & 0 & 8056 & 0 & 4069 & 0 & 1657 & 0 & 499 & 0 & 75 \\
75 & 0 & 435 & 0 & 1391 & 0 & 3319 & 0 & 6330 & 0 & 10010 & 0 & 16906 & 0 & 12406 & 0 & 7883 & 0 & 4231 & 0 & 1819 & 0 & 607 & 0 & 124 & 0 \\
0 & 75 & 0 & 435 & 0 & 1391 & 0 & 3319 & 0 & 6330 & 0 & 10010 & 0 & 16906 & 0 & 12406 & 0 & 7883 & 0 & 4231 & 0 & 1819 & 0 & 607 & 0 & 124 \\
124 & 0 & 607 & 0 & 1819 & 0 & 4231 & 0 & 7883 & 0 & 12406 & 0 & 16906 & 0 & 10010 & 0 & 6330 & 0 & 3319 & 0 & 1391 & 0 & 435 & 0 & 75 & 0 \\
0 & 124 & 0 & 607 & 0 & 1819 & 0 & 4231 & 0 & 7883 & 0 & 12406 & 0 & 16906 & 0 & 10010 & 0 & 6330 & 0 & 3319 & 0 & 1391 & 0 & 435 & 0 & 75 \\
75 & 0 & 499 & 0 & 1657 & 0 & 4069 & 0 & 8056 & 0 & 13208 & 0 & 18706 & 0 & 10606 & 0 & 4685 & 0 & 2505 & 0 & 1089 & 0 & 341 & 0 & 60 & 0 \\
0 & 75 & 0 & 499 & 0 & 1657 & 0 & 4069 & 0 & 8056 & 0 & 13208 & 0 & 18706 & 0 & 10606 & 0 & 4685 & 0 & 2505 & 0 & 1089 & 0 & 341 & 0 & 60 \\
60 & 0 & 375 & 0 & 1339 & 0 & 3457 & 0 & 7129 & 0 & 12316 & 0 & 18248 & 0 & 13666 & 0 & 6346 & 0 & 1625 & 0 & 705 & 0 & 229 & 0 & 41 & 0 \\
0 & 60 & 0 & 375 & 0 & 1339 & 0 & 3457 & 0 & 7129 & 0 & 12316 & 0 & 18248 & 0 & 13666 & 0 & 6346 & 0 & 1625 & 0 & 705 & 0 & 229 & 0 & 41 \\
41 & 0 & 265 & 0 & 949 & 0 & 2569 & 0 & 5548 & 0 & 10040 & 0 & 15760 & 0 & 14804 & 0 & 10755 & 0 & 4255 & 0 & 395 & 0 & 131 & 0 & 24 & 0 \\
0 & 41 & 0 & 265 & 0 & 949 & 0 & 2569 & 0 & 5548 & 0 & 10040 & 0 & 15760 & 0 & 14804 & 0 & 10755 & 0 & 4255 & 0 & 395 & 0 & 131 & 0 & 24 \\
24 & 0 & 161 & 0 & 601 & 0 & 1669 & 0 & 3793 & 0 & 7252 & 0 & 12056 & 0 & 13744 & 0 & 13100 & 0 & 9531 & 0 & 3535 & 0 & 59 & 0 & 11 & 0 \\
0 & 24 & 0 & 161 & 0 & 601 & 0 & 1669 & 0 & 3793 & 0 & 7252 & 0 & 12056 & 0 & 13744 & 0 & 13100 & 0 & 9531 & 0 & 3535 & 0 & 59 & 0 & 11 \\
11 & 0 & 79 & 0 & 315 & 0 & 931 & 0 & 2230 & 0 & 4574 & 0 & 8176 & 0 & 11132 & 0 & 12963 & 0 & 12539 & 0 & 9201 & 0 & 3381 & 0 & 4 & 0 \\
0 & 11 & 0 & 79 & 0 & 315 & 0 & 931 & 0 & 2230 & 0 & 4574 & 0 & 8176 & 0 & 11132 & 0 & 12963 & 0 & 12539 & 0 & 9201 & 0 & 3381 & 0 & 4 \\
4 & 0 & 31 & 0 & 135 & 0 & 435 & 0 & 1135 & 0 & 2514 & 0 & 4910 & 0 & 7840 & 0 & 10848 & 0 & 12759 & 0 & 12419 & 0 & 9145 & 0 & 3361 & 0
\end{bmatrix}$$

Figure 3.4. The transition probability matrix of the 16B9Q line code. The probabilities shown are multiplied by 65536.

$$\overset{\infty}{p} = Q^T \overset{\infty}{p} \qquad (3.29)$$

$$\sum_{i=1}^{J} \overset{\infty}{p}_i = 1 \qquad (3.30)$$

In mathematical terms, $\overset{\infty}{p}$ is the eigenvector of Q for the eigenvalue $\lambda = 1$ [51]. Equation (3.29) gives a set of 26 equations in 26 unknowns. However, only 25 of these equations are linearly independent. Therefore, Equation (3.30) is needed. The method of solution involves the substitution of (3.30) into (3.29) for one of the $\overset{\infty}{p}_i$s and solving the resulting set of linear equations. An equivalent procedure uses

$$\overset{\infty}{p} = \overset{\infty}{p}\, Q \qquad (3.31)$$

$$1 = \overset{\infty}{p}\, E \qquad (3.32)$$

where 1 is an N-dimensional row vector with all elements one, and E is a NxN matrix with all elements equal to one. Equation (3.31) is identical to Equation (3.29), while Equation (3.32) is simply Equation (3.30) written in matrix form. Adding (3.31) and (3.32) we obtain

$$\overset{\infty}{p} + 1 = \overset{\infty}{p}\, Q + \overset{\infty}{p}\, E$$

$$\overset{\infty}{p}(Q + E - I) = 1 \qquad (3.33)$$

giving

$$\overset{\infty}{p} = 1(Q + E - I)^{-1} \qquad (3.34)$$

where I is the identity matrix. Therefore, only one matrix addition, subtraction, inversion, and multiplication is needed. The computer program which carries out this calculation can be found in Appendix C (Program 2). Table 3.3 lists the absolute state probability vector $\overset{\infty}{p}$, and Figure 3.5 shows a plot of it.

Table 3.3. The Absolute State Probability Vector for the 16B9Q Line Code.

State	Probability
-13	0.5945373×10^{-3}
-12	0.5708554×10^{-3}
-11	0.3365472×10^{-2}
-10	0.3283899×10^{-2}
-9	0.1089395×10^{-1}
-8	0.1072183×10^{-1}
-7	0.2677235×10^{-1}
-6	0.2648606×10^{-1}
-5	0.5323558×10^{-1}
-4	0.5291646×10^{-1}
-3	0.8959498×10^{-1}
-2	0.8937288×10^{-1}
-1	$0.1321912 \times 10^{+0}$
0	$0.1321912 \times 10^{+0}$
1	0.8937288×10^{-1}
2	0.8959498×10^{-1}
3	0.5291646×10^{-1}
4	0.5323558×10^{-1}
5	0.2648606×10^{-1}
6	0.2677235×10^{-1}
7	0.1072183×10^{-1}
8	0.1089395×10^{-1}
9	0.3283899×10^{-2}
10	0.3365472×10^{-2}
11	0.5708554×10^{-3}
12	0.5945373×10^{-3}

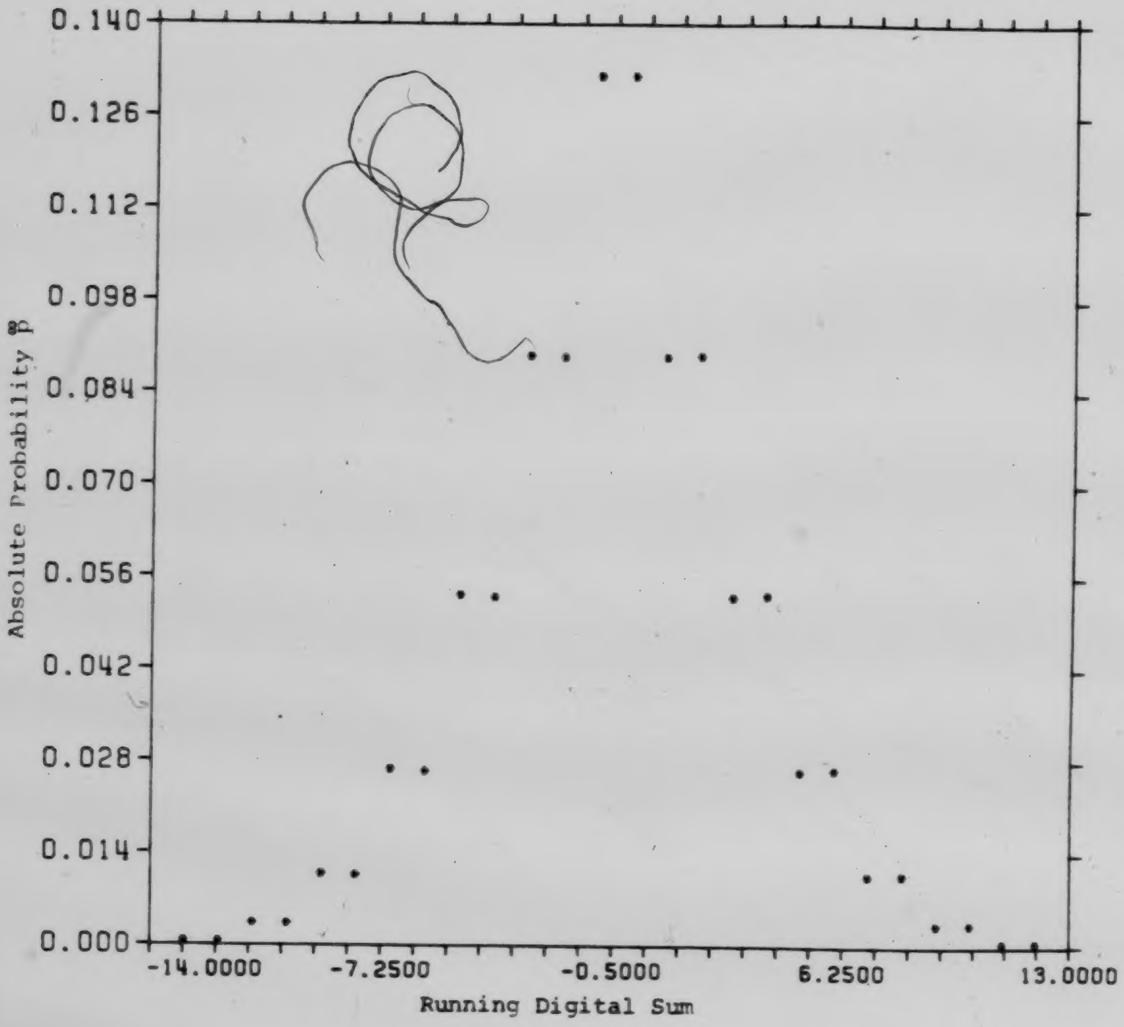


Figure 3.5. The absolute state probability vector $\overset{\infty}{p}$ of the 16B9Q line code.

3.4 Spectrum of the Coded Signal

The nth output word of the coder is denoted by $\mathbf{a}_n = (a_{n1}, a_{n2}, \ldots, a_{nN})$. Each symbol a_{ni} $(i=1,\ldots N)$ is transmitted on a standard pulse shape s(t) at intervals of duration T, so that t=NT is needed to transmit one N-length code word. The resulting PAM line signal can be expressed as

$$x(t) = \sum_{n=-\infty}^{\infty} \sum_{i=1}^{N} a_{ni}\, s[t-(nN + i-1)T]$$

$$= s(t) * \sum_{n=-\infty}^{\infty} \sum_{i=1}^{N} a_{ni}\, \delta[t-(nN+i-1)T]$$

$$= s(t) * y(t) \qquad (3.35)$$

where * denotes convolution. It is well known that convolution in the time domain corresponds to multiplication in the frequency domain. Therefore, we can find the power spectra of s(t) and Y(t) individually and multiply the results to get the overall power spectrum. Since s(t) is simply a standard pulse shape, we will concern ourselves only with finding the spectral density of Y(t).

The autocorrelation of the coded signal takes into account the length of the code word, as well as the position of the two correlated elements within the block, and is defined by

$$R_a^{ij}(k) = E\{a_{ni}\, a_{n+k,j}\} \qquad (3.36)$$

The autocorrelation and power spectrum of Y(t) are then

$$R_y(\tau) = \frac{1}{NT} \sum_i \sum_j \sum_k R_a^{ij}(k)\, \delta[\tau-(kN+j-i)T] \qquad (3.37)$$

$$P_y(f) = \frac{1}{NT} \sum_i \sum_j \sum_k R_a^{ij}(k) \exp[-j2\pi fT(kN+j-i)] \qquad (3.38)$$

This can be written in matrix notation by defining

$$R_k = E[a_n^T a_{n+k}] \tag{3.39}$$

$$V = (e^{j2\pi fT}, e^{j4\pi fT}, \ldots, e^{j2N\pi fT}) \tag{3.40}$$

Then, with V^* as the transposed conjugate of V,

$$P_y(f) = \frac{1}{NT} \sum_{k=-\infty}^{\infty} V R_k V^* e^{-j2\pi fkNT} \tag{3.41}$$

The Lebesgue decomposition [57] assures that any spectral density has two components: a continuous one and a discrete one. Equation (3.41) can thus be rearranged to obtain

$$P_y(f) = \frac{1}{N_T} \{X_c(f) + \frac{1}{NT} X_d(f) \sum \delta(f-kf_N)\} \tag{3.42}$$

where

$$X_c(f) = \sum_{k=-\infty}^{\infty} V(R_k - R_\infty)V^* e^{-j2\pi fkNT}$$

$$= V(R_o - R_\infty)V^* + 2 \operatorname{Re}\{\sum_{k=1}^{\infty} V(R_k - R_\infty)V^* e^{-j2\pi fkNT}\}$$

$$= V \Gamma_1 V^* + 2 \operatorname{Re}\{V \Gamma_2 V^*\}$$

$$= X_1(f) + 2 \operatorname{Re}\{X_2(f)\} \tag{3.43a}$$

$$\Gamma_1 = R_o - R_\infty \tag{3.43b}$$

$$\Gamma_2 = \sum_{k=1}^{\infty} (R_k - R_\infty) \exp[-j2\pi fkNT] \tag{3.43c}$$

and

$$X_d(f) = V\, R_\infty\, V^* \sum_{k=-\infty}^{\infty} e^{-j2\pi fkNT}$$

$$= V\, R_\infty\, V^* \sum_{k=-\infty}^{\infty} \delta(f-kNT) \tag{3.44}$$

The determination of the spectral density then requires the calculation of the sequence of correlation matrices $\mathbf{R}_k$, $k=0,1,2,\ldots$, its limit $\mathbf{R}_\infty = \lim_{K \to \infty} \mathbf{R}_K$ and the sum of the matrix series Γ_2.

The autocorrelation can be written in terms of the output matrix A (Equation (3.24)) as

$$R_k = E\{a_n^T a_{n+k}\} = A^T P_k A \tag{3.45}$$

where P_k is the joint probability matrix between the random vectors $\mathbf{a}_{n+k}$ and $\mathbf{a}_n$. Since A is composed of K submatrices $\mathbf{A}_u$, the probability matrix $\mathbf{P}_k$ must be partitioned into K^2 submatrices $\mathbf{P}_k^{uv}$ as follows

$$P_k = ||P_k^{uv}|| \quad (u,v=1,\ldots,K) \tag{3.46}$$

where the (i,j) entry of $\mathbf{P}_k^{uv}$ is given by

$$P_k^{uv}(i,j) = P[a_n = \alpha_{iu} \cap a_{n+k} = \alpha_{jv}] \tag{3.47}$$

Using (3.46) in (3.45) we get

$$R_k = \sum_{u=1}^{K} \sum_{v=1}^{K} A_u^T P_k^{uv} A_v \tag{3.48}$$

It can be proven [51] that the $\mathbf{P}_k^{uv}$ matrices are given by

$$P_o^{uv} = \delta_{uv} \; q_u \overset{\infty}{D} \tag{3.49}$$

$$P_k^{uv} = q_u q_v \overset{\infty}{D} E_u Q^{k-1}, \; k \geq 1 \tag{3.50}$$

where u,v=1,...,K, δ is the Kronecker delta, q_u are the input word probabilities, $\overset{\infty}{\mathbf{D}}$ is a diagonal matrix defined by

$$\overset{\infty}{D} = ||\overset{\infty}{P}_j \; \delta_{ij}|| \quad i,j=1,\ldots,J \tag{3.51}$$

$\mathbf{E}_u$ (u=1,...,K) are the transition matrices, and **Q** is the state transition probability matrix. Using (3.49) and (3.50) in (3.48) we obtain

$$R_o = \sum_{u=1}^{K} q_u \; A_u^T \; D \; A_u \tag{3.52}$$

$$R_k = C_1^T Q^{k-1} \; C_2 \quad k \geq 1 \tag{3.53}$$

where

$$C_1 = \sum_{u=1}^{K} q_u \; E_u^T \overset{\infty}{D} \; A_u \tag{3.54}$$

$$C_2 = \sum_{u=1}^{K} q_u \; A_u \tag{3.55}$$

By taking the limit as $k \rightarrow \infty$ of (3.53) we also get

$$R_{\infty} = C_1^T \overset{\infty}{Q} C_2 \tag{3.56}$$

where $\overset{\infty}{\bar{Q}}$ is the limiting transition probability matrix. The only term now left to be determined is Γ_2. Inserting (3.53) and (3.56) into (3.43 c) the expression

$$\Gamma_2 = C_1^T \Big[\sum_{k=1}^{\infty} w^{-k} (Q^{k-1} - \overset{\infty}{Q}) \Big] C_2 \tag{3.57}$$

is obtained, where $w = \exp(j2\pi fNT)$. The right hand side of (3.57) is a power matrix series which can be summed to yield [51]

$$\Gamma_2 = C_1^T [I - \overset{\infty}{Q}][wI - (Q - \overset{\infty}{Q})]^{-1} C_2 \tag{3.58}$$

Γ_2 is a complex matrix with no particular symmetries, and it is frequency dependent through w. Its evaluation requires the inversion of the matrix $[\lambda\, I - (Q - \overset{\infty}{Q})]$ and evaluation of the result for $\lambda = w$. This problem is well known in the theory of matrix functions [56] and a well defined method of solution exists. The matrix F is defined as $Q - \overset{\infty}{Q}$, and the inverted matrix is written as

$$(\lambda I - F)^{-1} = D(\lambda)^{-1}\, G \tag{3.59}$$

where $D(\lambda)$ is the characteristic polynomial of **F**, and **G** is the conjoint matrix of **F**. $D(\lambda)$ and **G** can be written as

$$D(\lambda) = |\lambda I - F| = \sum_{k=1}^{J} d_{J-k}\, \lambda^{k} \qquad d_o = 1,\ d_J = 0 \tag{3.60}$$

$$G = \sum_{k=0}^{J} G_{J-k}\, \lambda^{k-1} \qquad G_o = I,\ G_J = 0 \tag{3.61}$$

where the coefficients d_k and the matrix coefficients G_k are related by

$$G_K = F^K + d_1 F^{K-1} + \ldots + d_{K-1} F + d_K I \tag{3.62}$$

Therefore, as soon as $D(\lambda)$ is known, G is known. By expressing Γ_2 using (3.59) to (3.61) and performing some algebraic manipulation,

$$X_2(f) = V\,\Gamma_2\,V^* \tag{3.63}$$

becomes

$$X_2(f) = \frac{\displaystyle\sum_{s=-(N-1)}^{NJ-1} n_s \exp[js2\pi fT]}{\displaystyle\sum_{k=1}^{J} d_{J-k} \exp[jkN2\pi fT]} \tag{3.64}$$

where the coefficients n_k and d_k are frequency independent. The d_k and G_k coefficients can be calculated recursively using the equations [56]

$$d_k = -\frac{1}{k}\,\mathrm{trace}[FG_{k-1}]$$

$$G_k = FG_{k-1} + d_k I \qquad k=1,\ldots,J \tag{3.65}$$

with initial conditions $d_0=1$ and $G_0=I$. The computations can be verified by checking that $d_J=0$ and $G_J=0$. The numerator coefficients n_k are calculated by first finding

$$H_k = C_1^T(I-\overset{\infty}{Q})G_{J-k-1}\,C_2 \qquad k=0,\ldots,J-1 \tag{3.66}$$

each H_k being of dimension NxN, which then gives

$$n_{p+kN} = \sum_{i=1}^{N-p} H_k(i+p,i) + \sum_{i=1}^{p} H_{k+1}(i,i+N-p) \tag{3.67}$$

where $p=0,1,\ldots,N-1$, $k=-1,0,1,\ldots,J-1$, and $H_{-1}(i,j)=H_J(i,j)=0$.

To obtain the spectral density, it is necessary to take the real part of $X_2(f)$. This yields

$$\mathrm{Re}\{X_2(f)\} = \frac{\sum_{k=0}^{N(J+1)-1} N_k \cos(2\pi fkT)}{\sum_{k=0}^{J} D_k \cos(2\pi fkNT)} \tag{3.68}$$

where N_k and D_k are given by

$$N_{p+qN} = \sum_{h=o}^{J-q} d_{J-h}\, n_{(h+q)N+p} + d_{J-(h+q)}\, n_{hN-p} \tag{3.69}$$

$$D_k = \sum_{h=o}^{J-q} d_{J-h}\, d_{J-(h+q)} \tag{3.70}$$

with $p=0,1,\ldots,N-1$ and $q=0,\ldots,J$.

Now, since $\mathbf{R}_0$ and $\mathbf{R}_\infty$ are real symmetric matrices, the component $X_1(f)$ of $X_c(f)$ can be written as

$$X_1(f) = \sum_{k=o}^{N-1} \varepsilon_k\, \nu_k \cos(2\pi fkT) \tag{3.71}$$

and similarly, the discrete component $X_d(f)$ of the power spectrum can be expressed as

$$X_d(f) = \sum_{k=o}^{N-1} \varepsilon_k\, \mu_k \cos(2\pi fkT) \tag{3.72}$$

where

$\varepsilon_o = 1, \ \varepsilon_k = 2 \ (k \geq 1)$

$$\nu_k = \sum_{i=1}^{N-k} [R_o(i,i+k) - R_\infty(i,i+K)] \tag{3.73}$$

$$\mu_k = \sum_{i=1}^{N-k} R_\infty(i,\ i+k) \tag{3.74}$$

All the required coefficients have now been calculated and we can write

$$X_c(f) = \sum_{k=o}^{N-1} \varepsilon_k \nu_k \cos(2\pi fkT) + \frac{\sum\limits_{k=o}^{N(J+1)-1} \varepsilon_k N_k \cos(2\pi fkT)}{\sum\limits_{k=o}^{J-1} \varepsilon_k D_k \cos(2\pi fkNT)} \tag{3.75}$$

The total power spectral density is then the sum of the discrete and continuous parts

$$P_y(f) = X_c(f) + X_d(f) \tag{3.76}$$

The steps required in the calculation of the power spectral density of a line code are summarized below.

1. Determine the K code word submatrices $\mathbf{A}_u$.
2. Determine the transition matrices $\mathbf{E}_u$ which correspond to the $\mathbf{A}_u$.
3. Determine the input word probabilities q_u . If all input symbols are equally likely, then $q_u = 2^{-M}$.
4. Determine the state transition matrix $\overset{\infty}{\mathbf{Q}}$.
5. Determine the state probability eigenvector $\overset{\infty}{\mathbf{p}}$ by solving Equation (3.31). This also gives rise to $\overset{\infty}{\mathbf{Q}}$ and $\overset{\infty}{\mathbf{D}}$.
6. Determine $\mathbf{C}_1$ and $\mathbf{C}_2$ from (3.54) and (3.55).
7. Determine $\mathbf{R}_0$ from (3.52).
8. Determine $\mathbf{R}_\infty$ from (3.56).
9. Determine $\mathbf{G}_k$, $\mathbf{d}_k$, $\mathbf{H}_k$, $\mathbf{n}_k$, and finally $\mathbf{N}_k$ and $\mathbf{D}_k$ from (3.65), (3.66), (3.67), (3.69), and (3.70).
10. Finally calculate ν_k and μ_k from (3.73) and (3.74).

All that remains following completion of these ten steps is to substitute the coefficients into the sums X_c (f) and X_d (f).

3.5 Application to the 16B9Q Code

Two computer programs were written to implement the ten steps described in the previous section (Programs 3 and 4 in Appendix C).

An independent and equally likely input binary data stream is assumed in the analysis; therefore, the input word probabilities, q_u , are equal to 1/65536 (u=1,...,65536). The state transition probability matrix **Q**, the state probability eigenvector $\overset{\infty}{\mathbf{p}}$, the limiting transition matrix $\overset{\infty}{\mathbf{Q}}$, and the diagonal matrix $\overset{\infty}{\mathbf{D}}$ have already been obtained through the use of Program 2, as described in the previous section.

The determination of the code word submatrices $\mathbf{A}_u$, their corresponding transition matrices $\mathbf{E}_u$, and the matrices $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{R}_0$, is performed by Program 3 of Appendix C. It consists of a main program and four subroutines which create the input words, form $\mathbf{A}_u$ and $\mathbf{E}_u$, and perform the matrix multiplications $\mathbf{A}_u^T\mathbf{D}\mathbf{A}_u$, and $\mathbf{E}_u^T\mathbf{D}\mathbf{A}_u$. These operations must be performed for each of the 65536 input words, thus even though they are simple operations, a large amount of processing time is required. In this instance there is no short cut method available. The program takes approximately 770 seconds of central processing time to run if single precision variables are used, and approximately 800 seconds if the accuracy is increased to double precision (on the Amdahl 470/V8 mainframe computer). The resulting $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{R}_0$ matrices are used as input to Program 4, which calculates the remaining intermediate matrices and the final coefficients for the closed form expression of the continuous and discrete power spectral density. It was found that the precision of computation used did not result in any noticeable difference (i.e., the difference was in the fifth and less significant decimal places) to the final results. Therefore, the results of the single precision calculation for $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{R}_0$ are shown in Figures 3.6 and 3.7.

Program 4 consists of a main program, followed by seven subroutines. The matrices $\mathbf{G}_k$ and $\mathbf{H}_k$, the vectors $\mathbf{d}_k$ and $\mathbf{n}_k$, and the coefficients $\mathbf{N}_k$, $\mathbf{D}_k$, μ_k, and υ_k are computed here according to the appropriate equations. This computation requires only about 3 seconds of central processing time. Figures 3.8 and 3.9 display $\mathbf{N}_k$, $\mathbf{D}_k$, μ_k, and υ_k. These coefficients are read from left to right along the rows (i.e., the first entry in the first row is

$$
C_1 = \begin{bmatrix}
-0.84933E-03 & -0.84933E-03 & -0.84933E-03 & -0.84933E-03 & -0.12349E-02 & -0.12349E-02 & -0.12349E-02 & -0.12349E-02 & -0.89809E-03 \\
-0.21845E-03 & -0.21834E-03 & -0.21835E-03 & -0.21836E-03 & -0.12428E-02 & -0.12428E-02 & -0.12428E-02 & -0.12428E-02 & -0.48788E-03 \\
-0.28128E-02 & -0.28124E-02 & -0.28124E-02 & -0.28124E-02 & -0.80501E-02 & -0.80502E-02 & -0.80502E-02 & -0.80502E-02 & -0.24508E-02 \\
-0.78190E-03 & -0.78177E-03 & -0.78188E-03 & -0.78171E-03 & -0.80858E-02 & -0.80859E-02 & -0.80859E-02 & -0.80859E-02 & -0.18003E-02 \\
-0.74257E-02 & -0.74228E-02 & -0.74197E-02 & -0.74188E-02 & -0.15530E-01 & -0.15530E-01 & -0.15530E-01 & -0.15530E-01 & -0.80754E-02 \\
-0.18325E-02 & -0.18316E-02 & -0.18314E-02 & -0.18314E-02 & -0.15839E-01 & -0.15839E-01 & -0.15839E-01 & -0.15839E-01 & -0.39484E-02 \\
-0.15851E-01 & -0.15825E-01 & -0.15822E-01 & -0.15822E-01 & -0.27858E-01 & -0.27858E-01 & -0.27858E-01 & -0.27858E-01 & -0.11850E-01 \\
-0.18404E-02 & -0.18391E-02 & -0.18384E-02 & -0.18384E-02 & -0.28122E-01 & -0.28122E-01 & -0.28122E-01 & -0.28122E-01 & -0.84741E-02 \\
-0.25310E-01 & -0.25278E-01 & -0.25271E-01 & -0.25270E-01 & -0.35749E-01 & -0.35748E-01 & -0.35748E-01 & -0.35748E-01 & -0.18571E-01 \\
0.10318E-02 & 0.10334E-02 & 0.10328E-02 & 0.10330E-02 & -0.38283E-01 & -0.38283E-01 & -0.38283E-01 & -0.38283E-01 & -0.85717E-02 \\
-0.31784E-01 & -0.31717E-01 & -0.31712E-01 & -0.31711E-01 & -0.28802E-01 & -0.28802E-01 & -0.28802E-01 & -0.28802E-01 & -0.17472E-01 \\
0.10104E-01 & 0.10087E-01 & 0.10083E-01 & -0.10081E-01 & -0.29328E-01 & -0.29328E-01 & -0.29328E-01 & -0.29328E-01 & -0.14132E-02 \\
-0.28491E-01 & -0.28452E-01 & -0.28446E-01 & -0.28450E-01 & 0.39288E-03 & 0.39288E-03 & 0.39288E-03 & 0.39288E-03 & -0.10892E-01 \\
0.28491E-01 & 0.28452E-01 & 0.28446E-01 & 0.28450E-01 & -0.39288E-03 & -0.39288E-03 & -0.39288E-03 & -0.39288E-03 & 0.10892E-01 \\
-0.10104E-01 & -0.10087E-01 & -0.10083E-01 & -0.10081E-01 & 0.29328E-01 & 0.29328E-01 & 0.29328E-01 & 0.29328E-01 & 0.14132E-02 \\
0.31784E-01 & 0.31717E-01 & 0.31712E-01 & 0.31711E-01 & 0.28802E-01 & 0.28802E-01 & 0.28802E-01 & 0.28802E-01 & 0.17472E-01 \\
-0.10318E-02 & -0.10334E-02 & -0.10328E-02 & -0.10330E-02 & 0.38283E-01 & 0.38283E-01 & 0.38283E-01 & 0.38283E-01 & 0.85717E-02 \\
0.25310E-01 & 0.25278E-01 & 0.25271E-01 & 0.25270E-01 & 0.35749E-01 & 0.35748E-01 & 0.35748E-01 & 0.35748E-01 & 0.18571E-01 \\
0.18404E-02 & 0.18391E-02 & 0.18384E-02 & 0.18384E-02 & 0.28122E-01 & 0.28122E-01 & 0.28122E-01 & 0.28122E-01 & 0.84741E-02 \\
0.15851E-01 & 0.15825E-01 & 0.15822E-01 & 0.15822E-01 & 0.27858E-01 & 0.27858E-01 & 0.27858E-01 & 0.27858E-01 & 0.11850E-01 \\
0.18325E-02 & 0.18316E-02 & 0.18314E-02 & 0.18314E-02 & 0.15839E-01 & 0.15839E-01 & 0.15839E-01 & 0.15839E-01 & 0.39484E-02 \\
0.74257E-02 & 0.74228E-02 & 0.74197E-02 & 0.74188E-02 & 0.15530E-01 & 0.15530E-01 & 0.15530E-01 & 0.15530E-01 & 0.80754E-02 \\
0.78190E-03 & 0.78177E-03 & 0.78188E-03 & 0.78171E-03 & 0.80858E-02 & 0.80859E-02 & 0.80859E-02 & 0.80859E-02 & 0.18003E-02 \\
0.28128E-02 & 0.28124E-02 & 0.28124E-02 & 0.28124E-02 & 0.80501E-02 & 0.80502E-02 & 0.80502E-02 & 0.80502E-02 & 0.24508E-02 \\
0.21845E-03 & 0.21834E-03 & 0.21835E-03 & 0.21836E-03 & 0.12428E-02 & 0.12428E-02 & 0.12428E-02 & 0.12428E-02 & 0.48788E-03 \\
0.84933E-03 & 0.84933E-03 & 0.84933E-03 & 0.84933E-03 & 0.12349E-02 & 0.12349E-02 & 0.12349E-02 & 0.12349E-02 & 0.89809E-03
\end{bmatrix}
$$

$$
C_2 = \begin{bmatrix}
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.88387E+00 & 0.88387E+00 & 0.88387E+00 & 0.88387E+00 & 0.51428E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.85543E+00 & 0.85543E+00 & 0.85543E+00 & 0.85543E+00 & 0.50881E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.85543E+00 & 0.85543E+00 & 0.85543E+00 & 0.85543E+00 & 0.50881E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.77830E+00 & 0.77830E+00 & 0.77830E+00 & 0.77830E+00 & 0.49414E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.77830E+00 & 0.77830E+00 & 0.77830E+00 & 0.77830E+00 & 0.49414E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & 0.48181E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & 0.48181E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.32703E+00 & 0.32703E+00 & 0.32703E+00 & 0.32703E+00 & 0.40888E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.32703E+00 & 0.32703E+00 & 0.32703E+00 & 0.32703E+00 & 0.40888E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & -0.90454E-01 & -0.90454E-01 & -0.90454E-01 & -0.90454E-01 & 0.32829E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & -0.90454E-01 & -0.90454E-01 & -0.90454E-01 & -0.90454E-01 & 0.32829E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & 0.22559E+00 \\
0.89063E+00 & 0.89063E+00 & 0.89063E+00 & 0.89063E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & 0.22559E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & -0.22559E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & 0.81230E+00 & -0.22559E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & 0.90454E-01 & 0.90454E-01 & 0.90454E-01 & 0.90454E-01 & -0.32829E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & 0.90454E-01 & 0.90454E-01 & 0.90454E-01 & 0.90454E-01 & -0.32829E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.32703E+00 & -0.32703E+00 & -0.32703E+00 & -0.32703E+00 & -0.40888E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.32703E+00 & -0.32703E+00 & -0.32703E+00 & -0.32703E+00 & -0.40888E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & -0.48181E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & -0.81230E+00 & -0.48181E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.77830E+00 & -0.77830E+00 & -0.77830E+00 & -0.77830E+00 & -0.49414E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.77830E+00 & -0.77830E+00 & -0.77830E+00 & -0.77830E+00 & -0.49414E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.85543E+00 & -0.85543E+00 & -0.85543E+00 & -0.85543E+00 & -0.50881E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.85543E+00 & -0.85543E+00 & -0.85543E+00 & -0.85543E+00 & -0.50881E+00 \\
-0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.89063E+00 & -0.88387E+00 & -0.88387E+00 & -0.88387E+00 & -0.88387E+00 & -0.51428E+00
\end{bmatrix}
$$

Figure 3.6. The C_1 and C_2 matrices of the 16B9Q line code.

$$\begin{bmatrix}
0.50000E+01 & 0.57218E-07 & -0.23885E-08 & 0.57218E-07 & -0.38838E+00 & -0.38838E+00 & -0.38838E+00 & -0.38838E+00 & -0.78473E-01 \\
0.57218E-07 & 0.50000E+01 & -0.23885E-08 & 0.49788E-07 & -0.38820E+00 & -0.38820E+00 & -0.38820E+00 & -0.38820E+00 & -0.78801E-01 \\
-0.23885E-08 & -0.23885E-08 & 0.50000E+01 & 0.23891E-07 & -0.38828E+00 & -0.38828E+00 & -0.38828E+00 & -0.38828E+00 & -0.74813E-01 \\
0.57218E-07 & 0.49788E-07 & 0.23891E-07 & 0.50000E+01 & -0.38831E+00 & -0.38831E+00 & -0.38831E+00 & -0.38831E+00 & -0.74854E-01 \\
-0.38838E+00 & -0.38820E+00 & -0.38828E+00 & -0.38831E+00 & 0.50000E+01 & 0.17482E-08 & -0.58208E-10 & 0.0 & -0.84252E+00 \\
-0.38838E+00 & -0.38820E+00 & -0.38828E+00 & -0.38831E+00 & 0.17482E-08 & 0.50000E+01 & -0.58208E-10 & 0.0 & -0.84251E+00 \\
-0.38838E+00 & -0.38820E+00 & -0.38828E+00 & -0.38831E+00 & -0.58208E-10 & -0.58208E-10 & 0.50000E+01 & 0.0 & -0.84251E+00 \\
-0.38838E+00 & -0.38820E+00 & -0.38828E+00 & -0.38831E+00 & 0.0 & 0.0 & 0.0 & 0.50000E+01 & -0.84251E+00 \\
-0.78473E-01 & -0.78801E-01 & -0.74813E-01 & -0.74854E-01 & -0.84252E+00 & -0.84251E+00 & -0.84251E+00 & -0.84251E+00 & 0.67308E+01
\end{bmatrix}$$

Figure 3.7. The **R**, matrix of the 16B9Q line code.

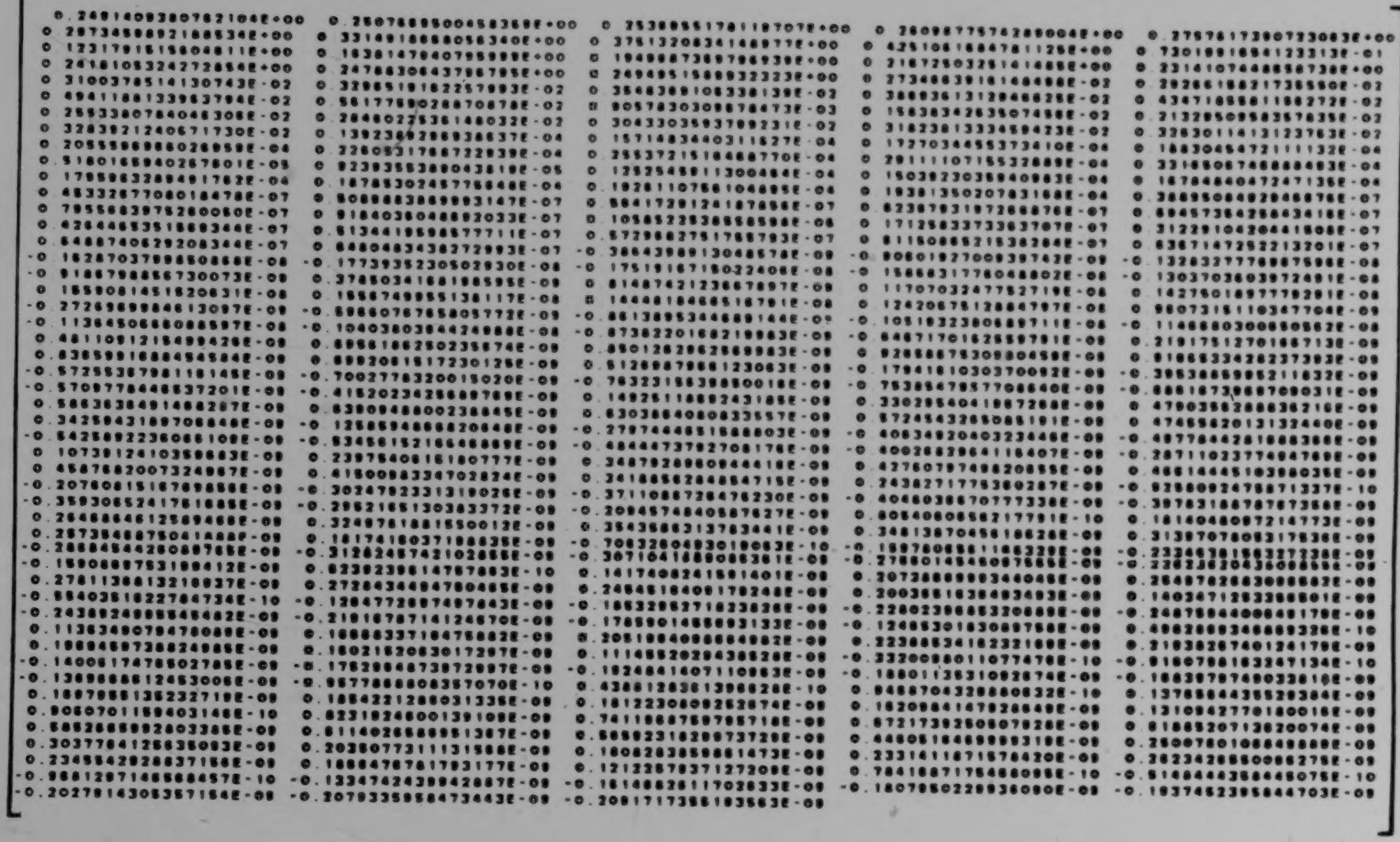


Figure 3.8. The numerator coefficients N_k of the 16B9Q line code.

```
  0.2000380780820480E+01   0.1012838306271233E+01   0.2838731638802828E-01   0.1220038805884083E-01  -0.8790031403843588E-03
 -0.8574038221284739E-03  -0.8356292261044788E-03  -0.8362483009952898E-03  -0.8741003086401232E-03  -0.8748414987343494E-03
 -0.2807289495889112E-03  -0.2814075938689804E-03  -0.4798303334570274E-04  -0.4910701398674910E-04   0.7027847782830811E-03
  0.7020748123250228E-03  -0.3498627511441884E-04  -0.3553990688835555E-04  -0.2737308585988822E-03  -0.2741023313498092E-03
 -0.8887403452054943E-03  -0.8869576390349141E-03  -0.8300010746753256E-03  -0.8301004800302004E-03  -0.1031826788834813E-02
 -0.1031862718336192E-02  -0.1171783818431049E-02
```

(a)

```
  0.4475143559800489E+02  -0.1031532736007311E+01  -0.1418057703487418E+01  -0.1808582670907808E+01  -0.2194107838387641E+01
 -0.1237380422384802E+01  -0.8498354549345341E+00  -0.4823104874848484E+00  -0.7478552003553551E-01
```

(b)

```
 -0.1379373295808578E-17   0.2587706123380874E-18  -0.2340818259405077E-18   0.1104458484852735E-17   0.1132845084175008E-18
 -0.7884373851482672E-18  -0.1227453117098982E-18  -0.7945423177101498E-18   0.4021828982025604E-18
```

(c)

Figure 3.9. The D_k, ν_k and μ_k coefficients of the 16B9Q line code.

(a) D_k

(b) ν_k

(c) μ_k

coefficient 1, while the first entry in the second row is coefficient 6, and so on).

Program 5 of Appendix C evaluates the power spectrum expression using these coefficients.

Since the μ_k coefficients are all zero, it is evident that the 16B9Q line code does not have a discrete component in its power spectrum. The continuous power spectrum of the code is plotted in Figure 3.10. This is the spectrum for Y(t), the train of coded, amplitude modulated delta pulses. In a practical system, the coded signal would be transmitted on some standard pulse shape, s(t). The observed power spectrum would therefore be the spectrum of s(t) multiplied by the spectrum of Y(t). If a rectangular pulse is used, its spectrum, $\text{sinc}^2(f)$, multiplies the curve shown in Figure 3.10. Thus, the higher frequencies are attenuated and the peak in the high frequency region disappears. The spectrum obtained is, in fact, that shown in Figure 5.9. It is evident that the curve of Figure 3.10 appears to be raised on a pedestal, whose value is the average power of the coded signal. The dc component is zero, however, since a non zero dc component would result in a discrete line at dc, and there is no such line. Further discussion, and a comparison of the theoretical and experimentally observed power spectra can be found in Chapter 5.

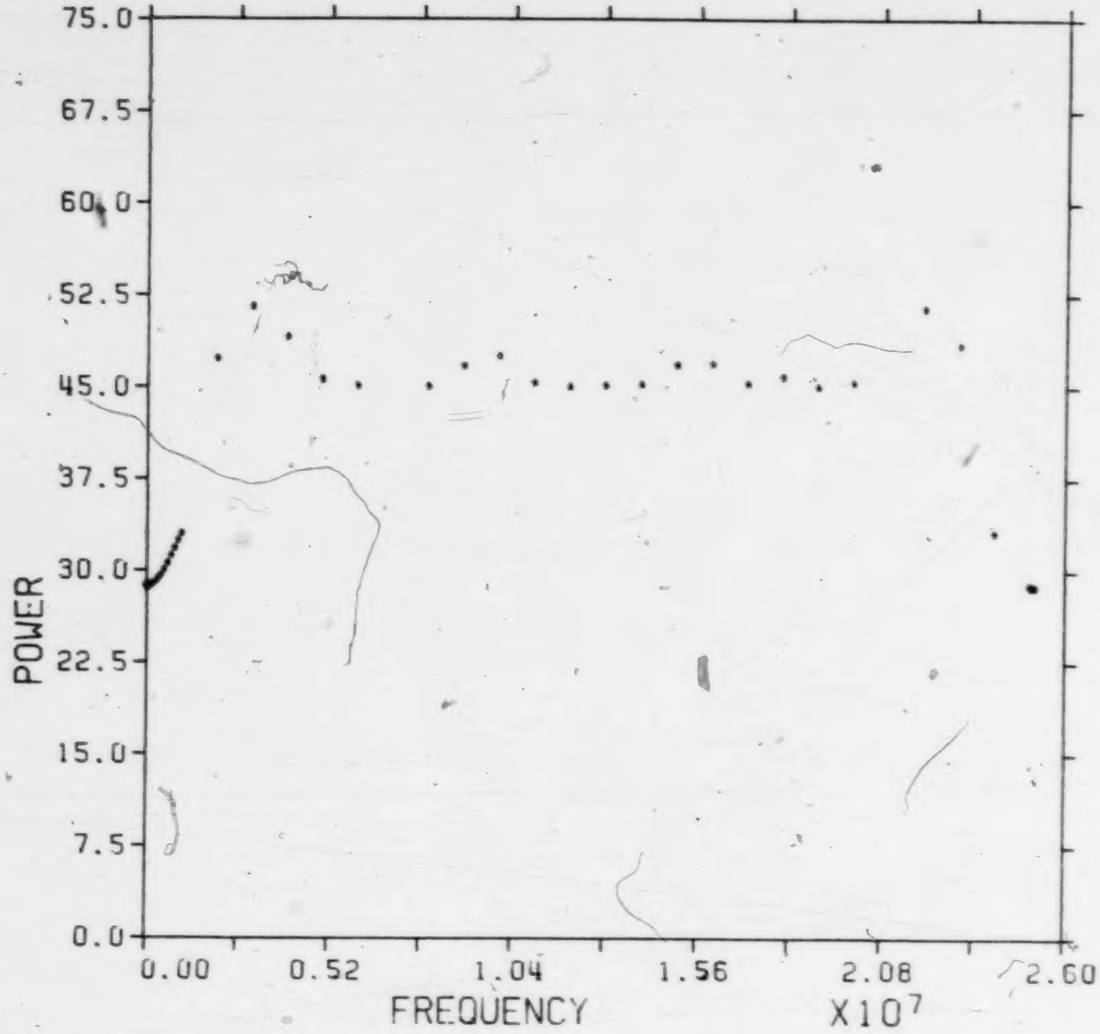


Figure 3.10. The analytically derived power spectrum of the 16B9Q line code.

CHAPTER 4

THE EXPERIMENTAL 16B9Q SYSTEM

An experimental optical fiber communication link was designed and constructed in order to verify the feasibility of employing the 16B9Q line code in a moderately high bit rate optical communication system. A block diagram of the system is shown in Figure 4.1. Table 4.1 explains the various symbols used in the figures of this chapter.

The coder accepts binary NRZ input data at 44.736 Mb/s, processes the signal according to the 16B9Q coding rules, and outputs a four level data stream at 25.164 Mb/s. This signal modulates a laser diode, which converts it to an optical four level signal. The output of the ILD enters the transmission channel, which is composed of multimode optical fiber. A variable optical attenuator installed in the middle of the link allows the amount of optical power reaching the photodetector to be accurately controlled. The preamplifier and main amplifier circuits amplify the received signal sufficiently so that it can be processed by the decoder. Upon converting the four level signal into a binary bit stream, the 16B9Q decoder performs the inverse operations of the coder, resulting in the original binary NRZ data at 44.736 Mb/s.

4.1 Coder

The 16B9Q line coder processes input binary data according to the coding rules and outputs a four level coded signal. It is composed of forty four integrated circuits (ICs) on six, six inch by four inch printed circuit boards (PCBs). The PCBs are placed in a card cage and are interconnected by coaxial cables on the backplane. They are double sided, with power traces routed on the ground plane side of the board and signal traces on the opposing side. Fifty ohm transmission line traces form the signal lines, which are terminated in fifty ohm pull down resistors. In order to minimize propagation delays (which can be significant at the high speeds at which the circuits operate) interconnections between boards and on the boards themselves are made as short as possible. Circuit operation at high speeds is accomplished by using emitter coupled logic (ECL) ICs.

Figure 4.2 shows a block diagram of the coder, while Figures 4.3 to 4.8 illustrate each of the six component circuit boards in greater detail. Table 4.2 lists the components of the coder. The block diagram depicts the basic data path of the incoming signal. Binary data is first

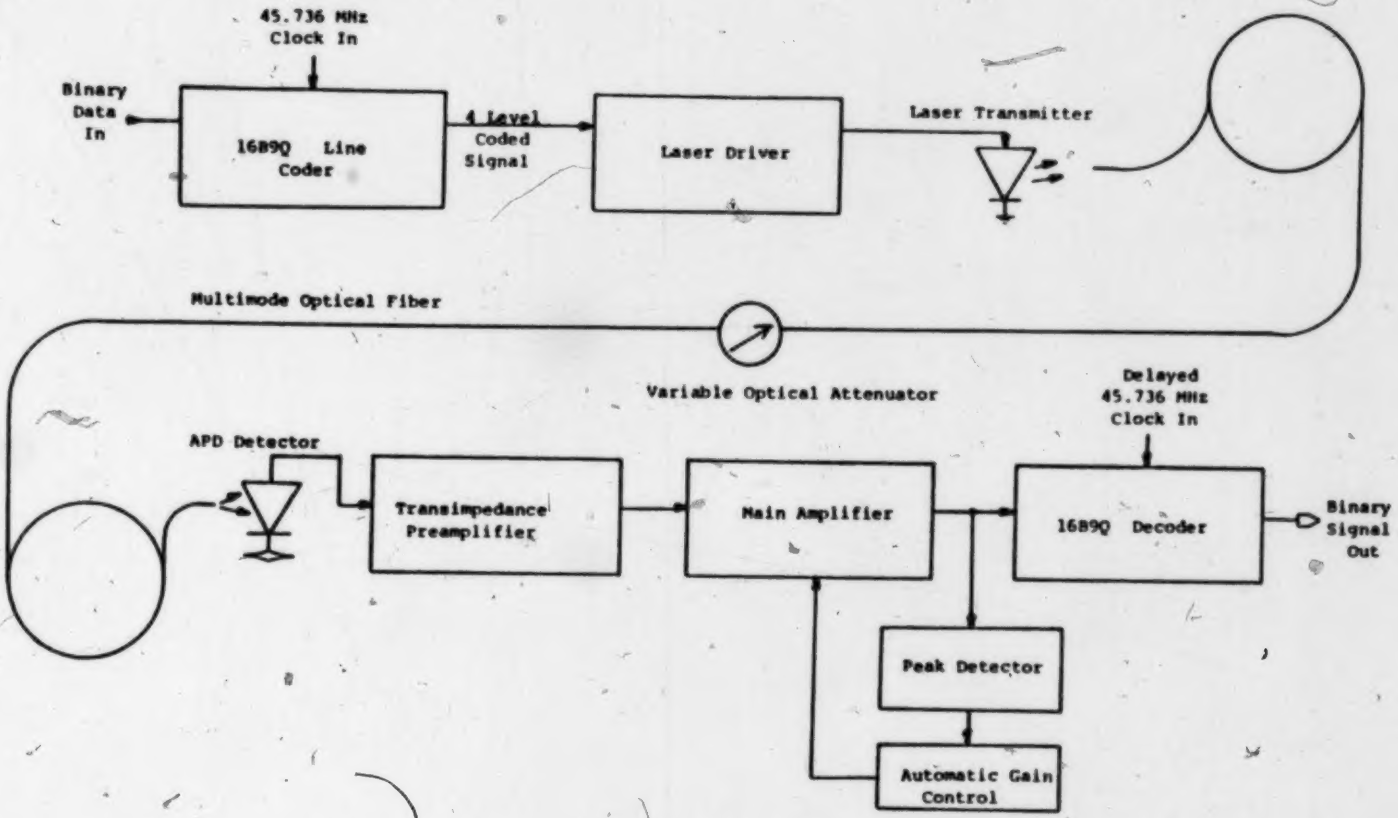


Figure 4.1. Block diagram of the experimental 16B9Q optical fiber system.

Table 4.1. Definition of Schematic Symbols.

⚲	-5.2 Volts	
□	+5 Volts	
◇	+6 Volts	
⏚	Ground	
⌇	51 Ω termination resistor, connected to -5.2 volts	
—▷|—	1N4148 diode used to create a 0.5 volt voltage drop	
—▷ƶ—	3.3 volt zener diode	
●—	signal entering the circuit board from another one	
◄—	signal leaving the circuit board for another one	
IC23 13	indicates the IC and pin number from which the signal originated or to which the signal is going. The direction of the arrow by which this symbol appears clearly indicates which of the above apply.	
pins 1, 16	connected to Ground	ON ALL ICs, UNLESS OTHERWISE INDICATED
pin 8	connected to -5.2 Volts	
All ICs	decoupled with 0.01 μF, connected between -5.2 Volts and Ground	

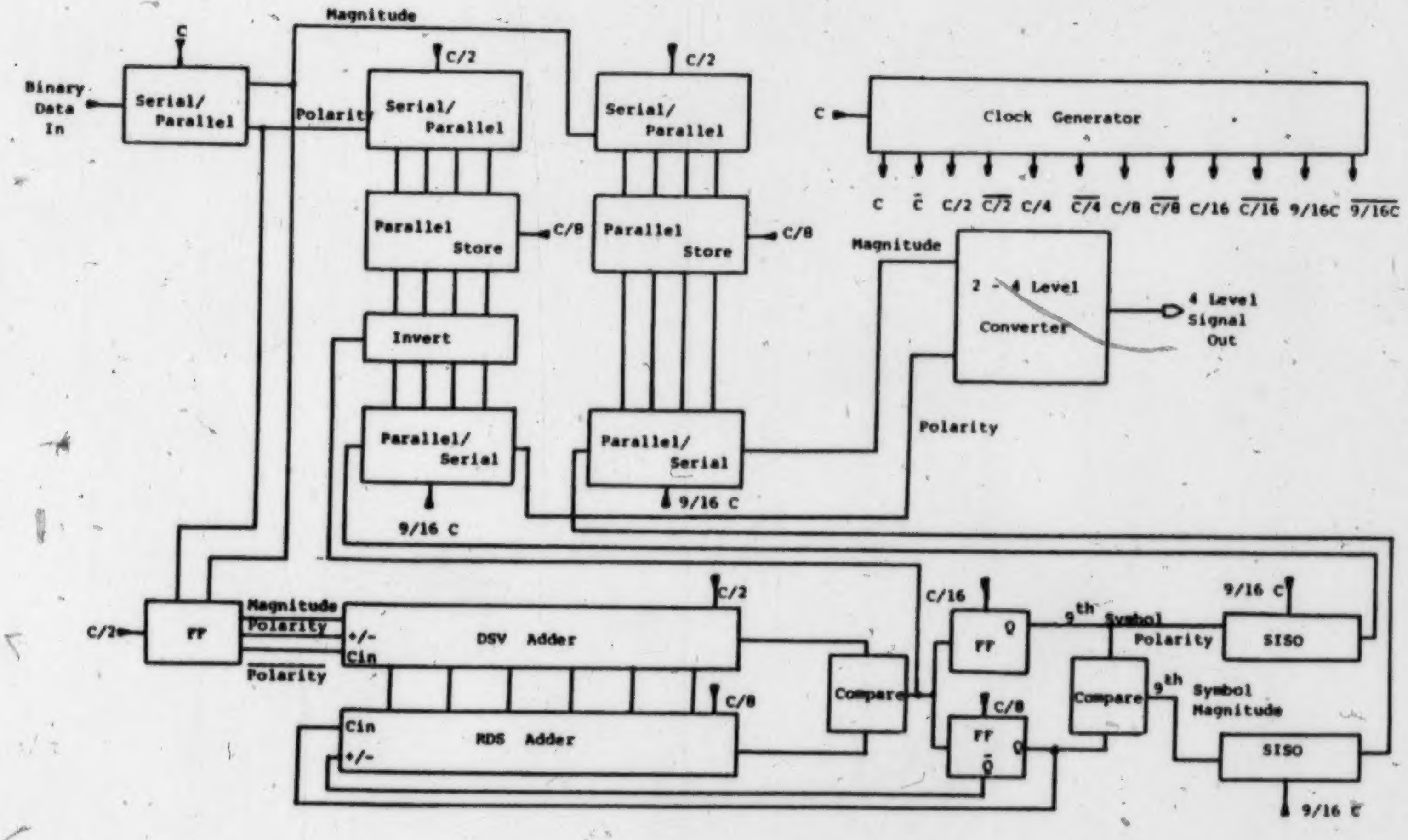


Figure 4:2. Block diagram of the 16B9Q line coder.

Table 4.2. List of Coder Components.

Component Number	Component
IC 1,3	MC10101
IC 40	MC10103
IC 21	MC10104
IC 33,37	MC10107
IC 20,39	MC10109
IC 14	MC10114
IC 10	MC10124
IC 7,43	MC10125
IC 19,23,25,27,34,35,38,41	MC10131
IC 12,13,15,16,17,18,39,42	MC10H141
IC 2	MC10154
IC 42	MC10172
IC 4,5,6,28,32	MC10H176
IC 22,24,26,29,30,31	MC10H180
IC 8	NE564
IC 9	74F161
IC 44	74ALS38
R 1,2,6	1 k Ω
R 3	9.1 k Ω
R 4	2.7 k Ω
R 5	100 Ω
C 1,3,4,9	0.01 µF
C 2,6	0.1 µF
C 5	20 pF varicap
C 7,8	820 pF
P 1,2,3,4,5	10 k Ω potentiometers

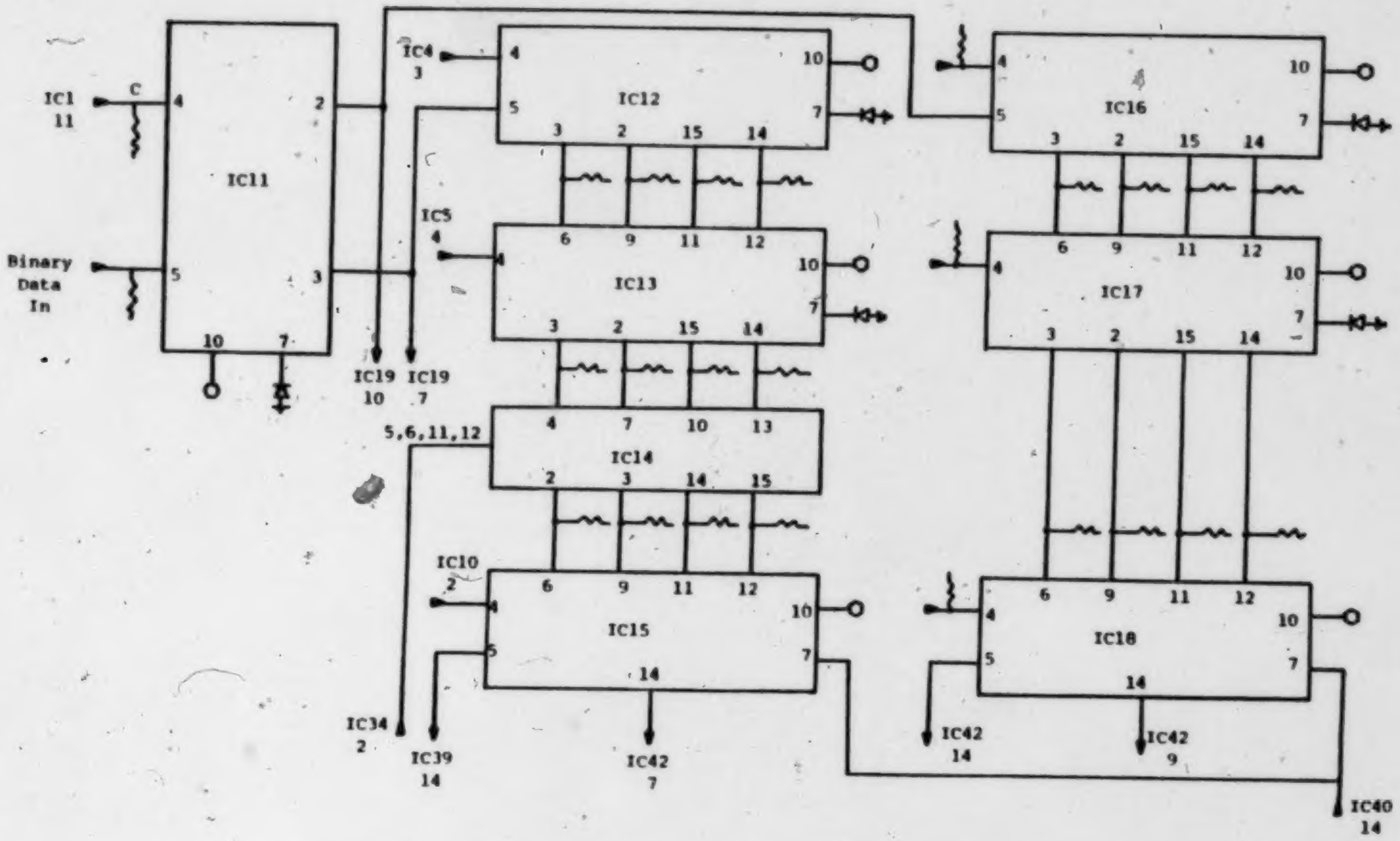


Figure 4.4. The data acceptance, storage, and inversion circuit of the coder.

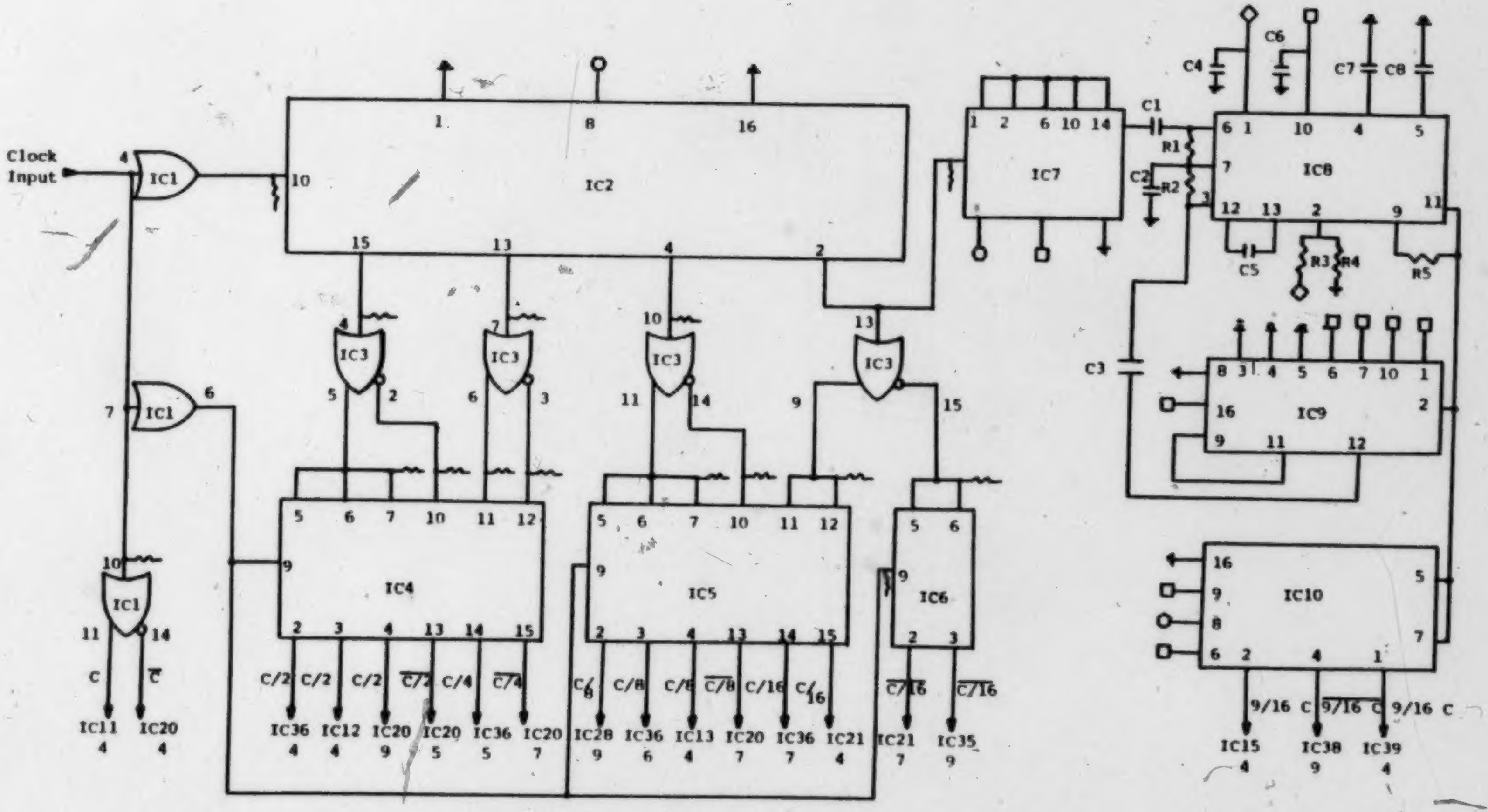


Figure 4.3. The coder clock generating circuit.

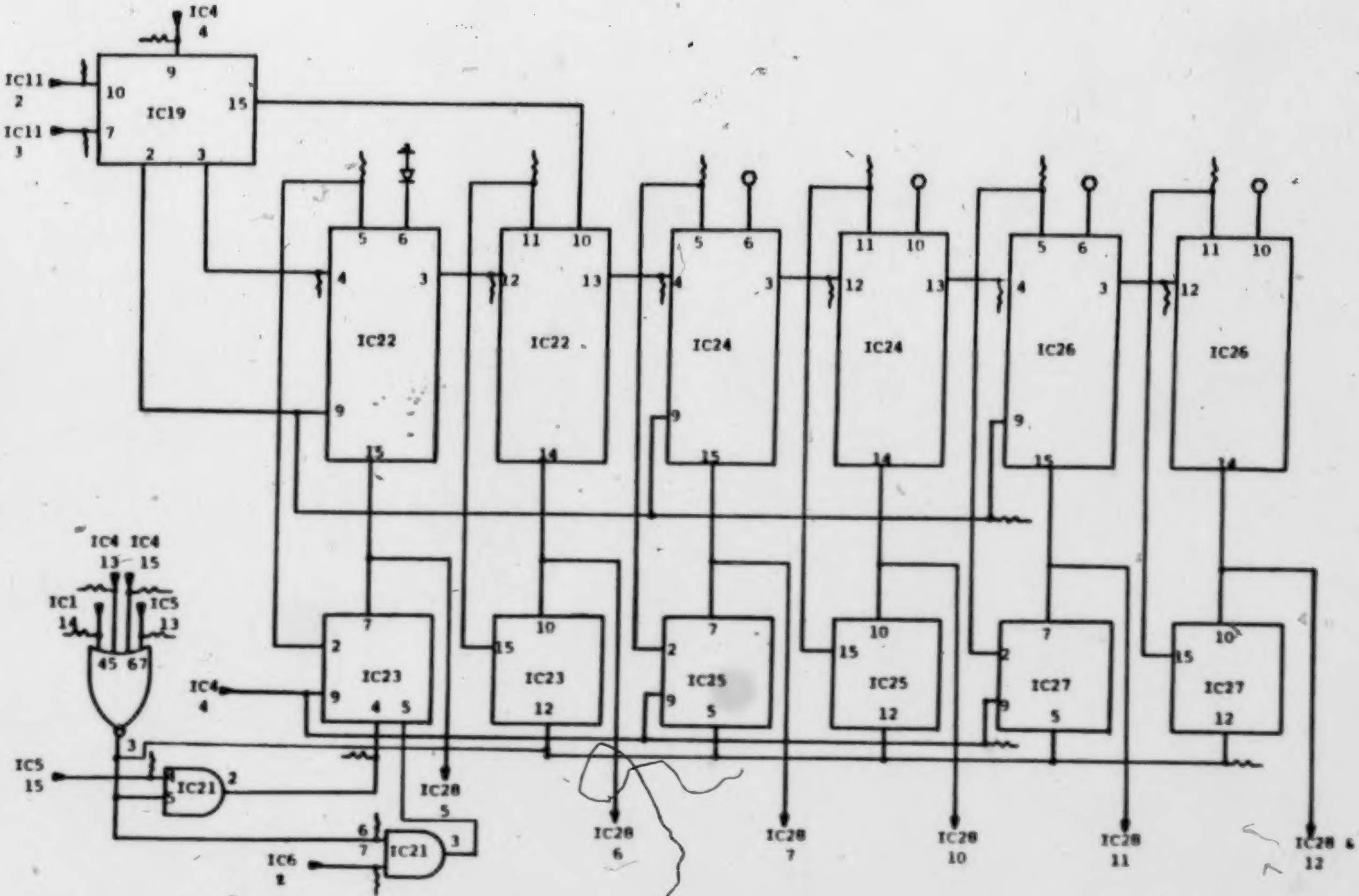


Figure 4.5. The coder DSV adder circuit.

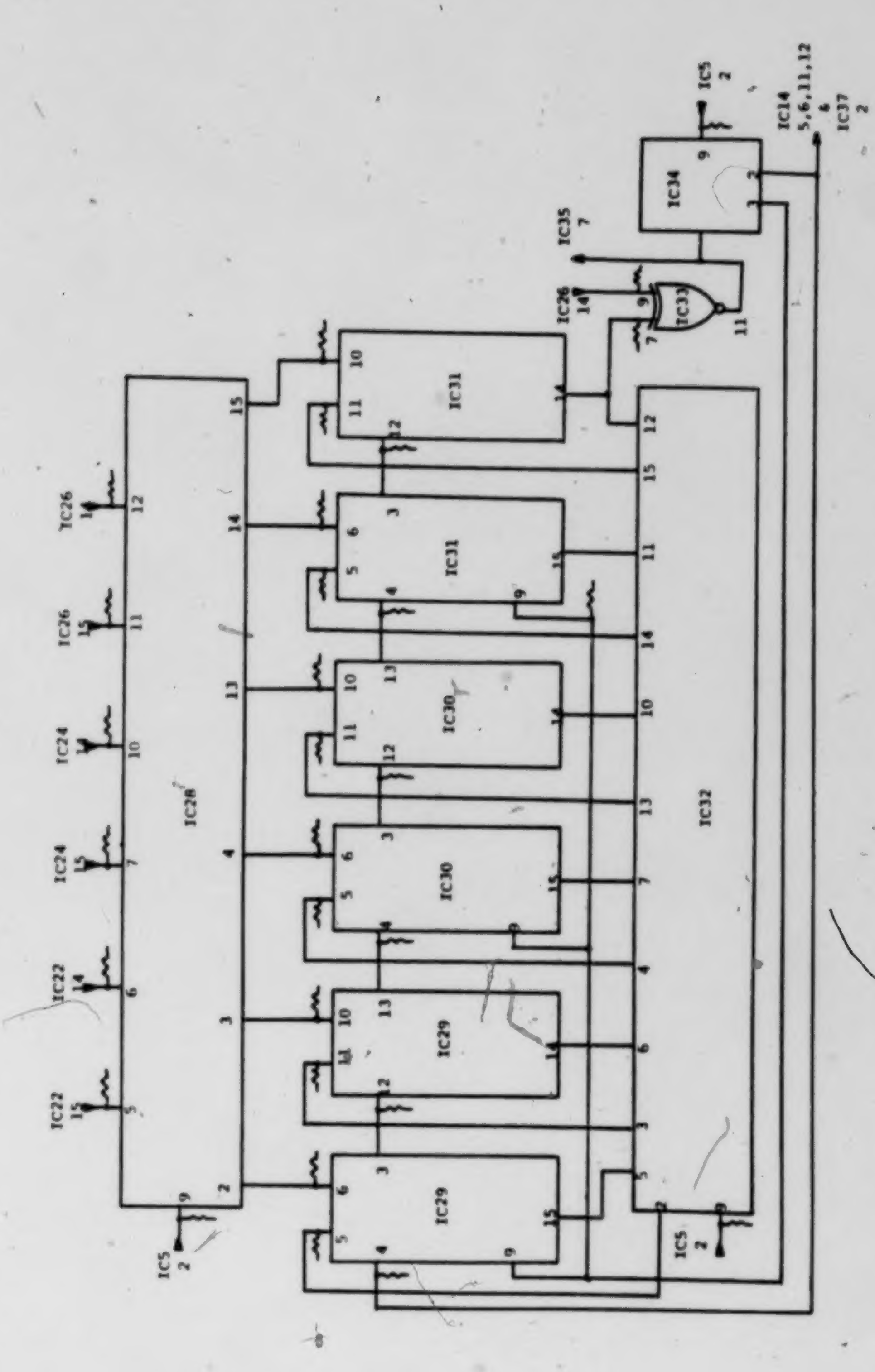


Figure 4.6. The coder RDS adder circuit.

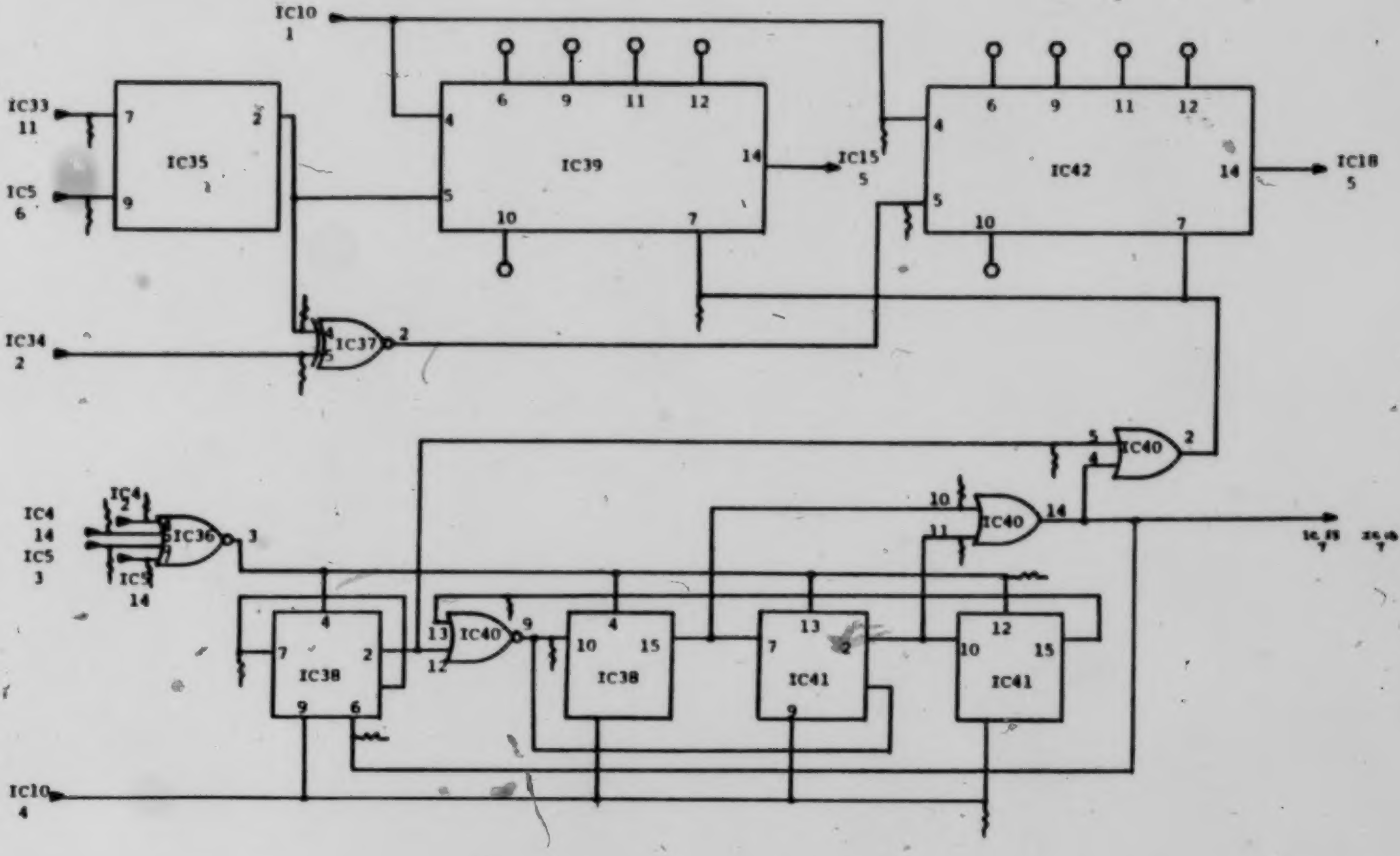


Figure 4.7. The coder ninth symbol generator circuit.

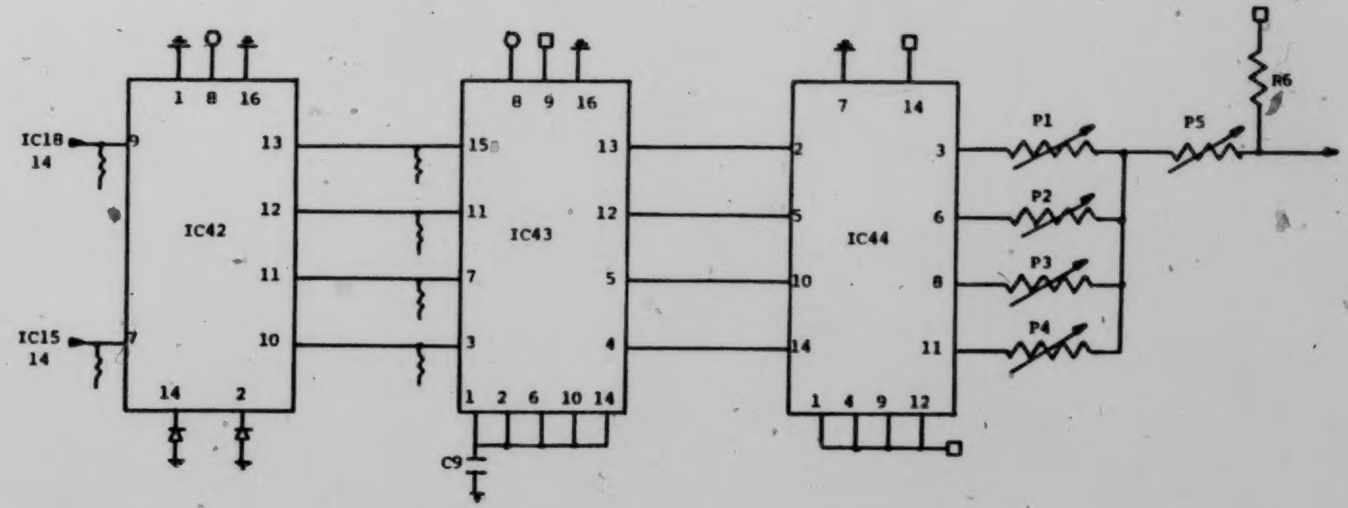


Figure 4.8. The binary to four level converter circuit.

converted into two parallel bit streams, then stored in shift registers while the DSV, RDS, and ninth symbol calculations take place, then inverted as needed, and finally shifted out. All processing of the data takes place in binary form. It is only at the final step before being sent to the transmitter that a four level signal is created.

The coder clock generator shown in Figure 4.3 creates the various clock rates needed to operate the coder. IC1 buffers the clock input to allow an increased fanout, while IC2, a binary counter, divides the clock input by two, four, eight, and sixteen. These four outputs pass through buffer IC3, are retimed by D flip flops (IC4,5,6), and are then available to drive various coder circuits. As the final output of the coder will be transmitted at a frequency of 9/16 times the input clock rate, a high frequency phase locked loop (PLL) is needed to generate this 9/16 clock signal. The input clock divided by sixteen is first converted to TTL voltage levels from ECL levels by IC7, since the PLL, IC9, is a TTL device. Capacitor C5, a variable capacitor, is adjusted to set the free running frequency of the PLL to a value close to the desired output frequency. Capacitors C7 and C8, together with an internal resistor, act as a low pass filter, while resistors R3 and R4 control the gain of the PLL. IC9, a TTL hexadecimal counter in the feedback loop of the PLL circuit, is set to divide by nine. The divided by sixteen input signal enters the PLL and is compared with the output of the voltage controlled oscillator (VCO) divided by nine. The resulting PLL output is the desired 9/16 clock signal, which is converted back to ECL levels by IC10.

Figure 4.4 depicts the circuit responsible for accepting, storing, inverting (as needed), and shifting data through the coder. IC11 acts as a serial to parallel converter. The input data signal is split into two parallel bit streams, the odd numbered bits representing the magnitude of the four level output signal, and the even numbered ones representing the polarity. As these two bit streams are shifted into IC12 and IC16, respectively, at a rate of C/2 (where C is the clock rate), they also go to the DSV adder (Figure 4.5) where the DSV and subsequently RDS (Figure 4.6) are computed. In order to allow sufficient time for these calculations to take place, after four C/2 periods the bits stored in IC12 and IC16 are parallel loaded into IC13 and IC17, where they are stored for another four C/2 periods. Shortly after the parallel load occurs the signal indicating whether an inversion is needed arrives at IC14 (exclusive OR gates). During an inversion the polarity changes while the magnitude remains the same; therefore, it is only the polarity bits that go through inverters. A short time following the assertion of the inversion signal at IC14, all eight bits are present (four magnitude and four polarity) with their coded values (the magnitude bits are unchanged; the polarity bits may be inverted) at the parallel inputs of the final set of shift registers, IC15 and IC18. These shift registers operate at a rate of (9/16)C. On the leading edge of the C/16 pulse their parallel inputs are loaded with the first word of the two word frame. They shift right three times, parallel load again (the second word), and shift right four times. On the second parallel load, the ninth symbol polarity and magnitude bits are present at the serial inputs of the shift registers. In the following four shifts, they are moved through the registers and are thus inserted into the outgoing bit stream.

Figures 4.5 and 4.6 illustrate the DSV and RDS adders. It is in these adders that these sums are calculated and stored, enabling the decision of whether to invert or not to be made. Both the DSV and RDS adders are constructed as ripple through adders. DSV additions occur serially, while the RDS is calculated with a parallel addition. Two's complement arithmetic is used.

In the split block coder the maximum instantaneous RDS that can be attained is $(9/4)n+3$. The coder must be able to obtain this sum without the occurrence of an overflow condition. Therefore the order of adder needed for a 2nB/(n+1)Q line code is $[\mathrm{Int}(\log_2\lceil(9/4)n+3\rceil)+1]$. For the 16B9Q line code this expression evaluates to $5+1=6$, that is, a six bit adder. A five bit adder would hold the value of the maximum RDS, but as it is essential to know the sign of the RDS (and DSV), an extra bit is needed for use as a sign bit. ICs 22,24,26,29,30, and 31 are dual one bit adders. Since these adders are not synchronous, external provisions must be made in order to assure that additions occur only when wanted and that they involve the proper operands. This is accomplished by storing the control signals and the numbers to be added in flip flops (ICs 19,23,25,27,28,32,34) which are clocked at appropriate times.

Magnitude and polarity bits are clocked into IC19 of the DSV adder at a rate of C/2. The polarity bits determine whether addition or subraction occurs. For subtraction, a binary one is fed into the carry in (Cin) input of the least significant bit (LSB) to create a 2's complement number. The adder IC only inverts the number to be subtracted; therefore, this extra step is required. The magnitude bits determine the magnitude of the number to be added or subtracted. Since this magnitude will always be a 1 or a 3, the LSB of the first operand is tied high. The magnitude bits toggle the next most significant bit, and the remaining four bits of the first operand are tied low. The result of an addition is stored in flip flops (ICs 23,25,27), which also feed it back as the second operand of the adders, for the next addition. The DSV is initialized at the beginning of the first four bit word in the two word frame to -2, and at the beginning of the second word to -1. This is accomplished by setting and resetting the appropriate flip flops at the appropriate times. Every C/8 the five most significant bits of the flip flops are set to ones. On C/16 the LSB is reset to a zero (thus creating a -2), while on C/16 it is set to a 1 (thus creating a -1). The set and reset pulses are created by IC20 and gated through at the appropriate times by IC21. After these initializations, the additions proceed as described above. Using appropriate logic families, an addition can be completed in 10 ns. or less, allowing the adder to operate at rates exceeding 100 Mhz.

The RDS is determined by adding the DSV of the four bit word just calculated to the previous RDS, using a parallel addition. That is, every C/8 the entire DSV 2's complement number is clocked into the second operand of the RDS adder (by IC28), while at the same time the previous RDS 2's complement number is clocked into the first operand (by IC32). The addition proceeds as usual, and as stated earlier, takes only 10 ns. Following this, the RDS adder is idle until the next C/8 pulse.

The RDS calculated here is the correct value of RDS, reflecting the actual values of the encoded bits sent out of the coder. That is, all inversions of four bit words are taken into account, as well as the value of the ninth symbol (before it has been created). This unique feature allows a more than sufficient amount of time for processing the signal.

An inversion is determined by comparing the polarity of the previous RDS to the polarity of the current DSV. This is accomplished by feeding the sign bits (the most significant bits) of the DSV and RDS adders into an exclusive NOR gate (IC33). The output of IC33 is sent to IC14 where the inversion, if indicated, occurs. It is also stored in IC34 until the next RDS addition when it determines whether addition or subraction of the RDS will occur. If a block of four bits is inverted, the DSV of the word that is actually transmitted is the inverse of the DSV originally calculated. Therefore, to obtain the correct RDS, this DSV must be subtracted from the current RDS. This process occurs in the RDS adder. If a word is to be inverted before transmission, the DSV of that word is subtracted from the RDS to obtain the correct current value of RDS. The subtraction is again a 2's complement operation, and is determined by the result of the comparison of the two polarities (this is the same signal that performs the inversion).

The initialization of the DSV adder to -2 and to -1 at the beginning of the first and second words allows the value of the ninth symbol to be correctly added to the RDS, before this symbol has been created. The initializations and their appropriate inversions determine the DSV of the ninth symbol, as seen from Table 4.3.

Figure 4.7 displays the ninth symbol generator. The ninth symbol consists of a magnitude bit and a polarity bit, and is created immediately after the DSV of the second word in the frame has been added to the RDS to obtain the RDS at the end of the current frame. As can be seen from Table 2.4, the value of the polarity bit is determined by the inversion decision performed on the first word. If the first word was inverted, the polarity is positive (a binary one), and if it was not inverted, the polarity is negative (a binary zero). This decision is stored in IC35 and is output at the appropriate time. The magnitude of the ninth symbol can be determined by performing an exclusive NOR of the inversion decisions performed on the first and second words. This operation is performed by IC37. Upon creation of the magnitude bit (after the second word DSV is added to the RDS), the entire data frame (two streams of eight bits each) is stored in four shift registers (ICs 13,15,17,18) and has not yet been transmitted out of the coder. The ninth symbol must therefore be stored in order for the coder to be able to insert it in its proper slot in the outgoing bit stream. This happens in the following manner. On C/16 the ninth symbol shift registers (IC39 and IC42) are parallel loaded with zeroes. At this time the ninth symbol is present at the serial inputs of these shift registers, which are serially connected to the final shift registers (IC15 and IC18). After the parallel load, the registers (IC39, IC42) shift right eight times, clocking out four zeros followed by the ninth symbol. The four zeros are shifted into the final registers, but are not moved out of them, since the second word of the frame is loaded into IC15 and IC18 on the clock pulse that otherwise would cause a zero to appear at their serial outputs. On that clock pulse also, the ninth symbol has been clocked out of the ninth symbol shift registers and is present at the serial inputs to IC15 and IC18. Thus, it is inserted into its proper place in the output stream. As soon as the ninth symbol appears at the serial outputs, another parallel load occurs, preventing the bits that followed the ninth symbol from being transmitted. ICs 36,38,40, and 41 form a ÷4/÷5 and a ÷9 counter, which is used to control the operation of ICs 39,41,15, and 18. The ÷9 output causes ICs 39 and 42 to parallel load and shift right eight times, while the ÷4/÷5 output causes ICs 15

Table 4.3. Creation of the 9th Symbol in the 16B9Q Line Code.

First Word	Second Word	Result on First Word	Result on Second Word	9th Symbol DSV
Not inverted	Not inverted	-2	-1	-3
Not inverted	Inverted	-2	+1	-1
Inverted	Not Inverted	+2	-1	+1
Inverted	Inverted	+2	+1	+3

and 18 to parallel load, shift right three times, parallel load, and shift right four times.

The final stage in the 16B9Q line coder is the conversion of the binary signal into a four level signal. This is accomplished by the circuit shown in Figure 4.8. The parallel binary bit streams from ICs 15 and 18 are inputs to IC42, a two to four line decoder. This IC has four outputs, one of which is high at any time. The particular output which is high is determined by the combination present at the two inputs. The four outputs of IC42 are converted to TTL levels by IC43 and to open collector signals by IC44. Then they are weighted with different values of resistance (P1 to P4), to create one of four voltage levels, and summed by wire ORing. The differential weighting and summing of the four signals creates one four level signal. The open collector outputs of IC44 enable the independent adjustment of these levels. That is, one level can be adjusted without affecting the others in any way. Potentiometer P5 is used to adjust the final output signal, to meet the requirements of the transmitter.

4.2 Transmitter

The optical transmitter used in the experimental system is a General Optronics short wavelength (.82μm) single longitudinal mode laser diode, which has a laser drive circuit in the same package (model GO-ANA). The laser can be modulated over a frequency range of 20 Hz to 1.25 GHz, has a typical output power of 2 mW, and has a fairly linear output power characteristic. The drive circuit incorporates automatic power and temperature control and a performance warning alarm. If the laser is biased just below threshold, as in the case of binary transmission, the time needed to reach each of the three non zero light levels will be different, and the delay will increase as the level does. This limits the attainable modulation rate and causes timing and decoding problems at the receiver. In order to avoid these problems, the laser is adjusted so that its bias point is in the center of its linear operating region, allowing it to be modulated above and below this point. The unequal turn on and turn off delays are therefore reduced and transmission of an electrically bipolar signal is possible without additional level translation. The input multilevel electrical signal is capacitively coupled through a fifty ohm connector to the laser driver and its amplitude is adjusted to 0.4 V peak to peak.

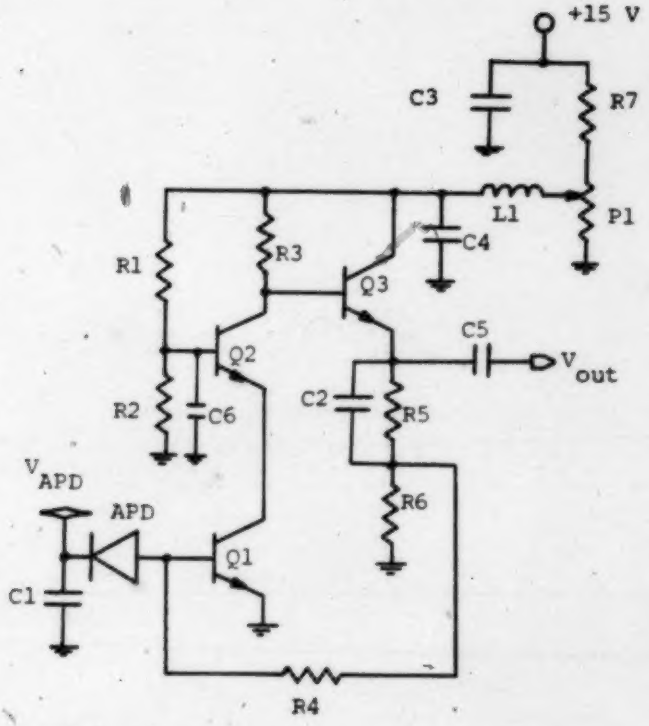
4.3 Fiber

The experimental system utilizes graded index, multimode fiber. The fiber used to verify the dispersion immunity property of the 16B9Q line code was Siecor Super-Fat Fiber Cable #155. It has a high numerical index of 0.4, an attentuation of 35 dB/km at 830 nm, and a bandwidth of 5 MHz-km . Approximately 200 meters of this fiber are needed in order to transmit data at a rate of 25.164 Mb/s. The remaining portions of the experimental work employed a much higher quality Northern Telecom fiber, characterized by a numerical aperture of 0.2, an average attenuation of 2.75 dB/km, and a dispersion of 1.5 ns/km. The system was tested with five kilometers of this fiber in place. A variable optical attenuator placed in series with the fiber link enabled precise control of the optical power falling upon the receiver to be obtained.

4.4 Receiver

The optical receiver is composed of an APD photodetector followed by a transimpedance preamplifier and a main amplifier incorporating automatic gain control. The APD is a RCA C30908E with a quantum efficiency of 77% at 830 nm, a mean gain of 150, and an ionization ratio of 0.02. The preamplifier circuit is shown in Figure 4.9. The cascode transistor pair forming the input stage exhibits a good high frequency response, high current gain, and low noise. A common collector output stage is added as a buffer, to enable the use of a following external 50 ohm amplifier. Voltage shunt feedback by means of resistor R4 lowers the effective input resistance of the circuit while keeping the actual physical resistance high. Thus, the preamplifier bandwidth is increased without an increase in thermal spot noise. The feedback also increases the dynamic range of the preamplifier.

The original design procedure for this circuit was developed by El-Diwany et al. [43] and was used by Kelly [44] to implement a preamplifier with a bandwidth of 10 MHz. The same procedure was followed here to implement a 30 MHz bandwidth preamplifier. The design calculations can be found in Appendix A. Final circuit analysis was performed using SCAMPER, an analog circuit analysis computer program developed by Bell-Northern Research. Figure 4.10 is a graph of the expected preamplifier frequency response, as determined by the computer analysis.



R1,2 = 2.7 kΩ
R3 = 1.5 kΩ
R4 = 6.9 kΩ
R5 = 300Ω
R6 = 75Ω
R7 = 5.0 kΩ
R8 = 220Ω
L1 = 2 μH

C1,3,4 = 0.1μF
C2 = 22μF/16V
C5 = 0.01μF
Q1,2,3 = BFR90
C6 = 0.1μF
P1 = 10 kΩ

Figure 4.9. The experimental optical detector/preamplifier circuit.

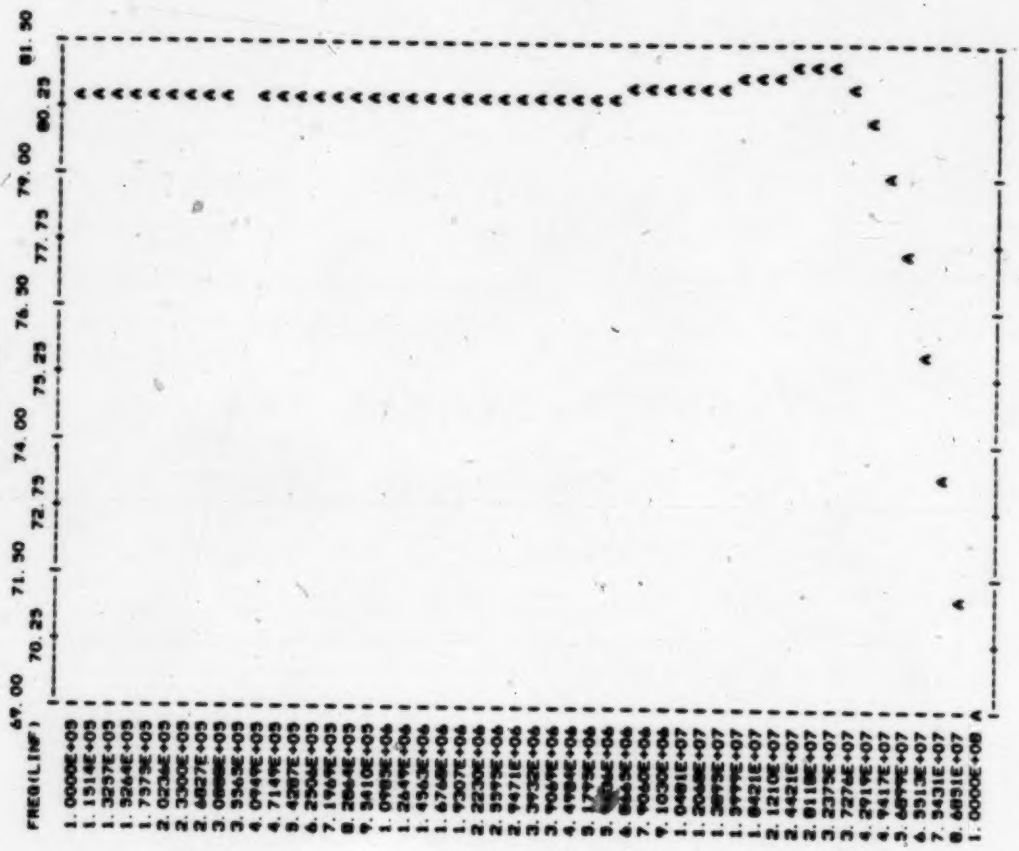


Figure 4.10. Expected frequency response of the transimpedance preamplifier.

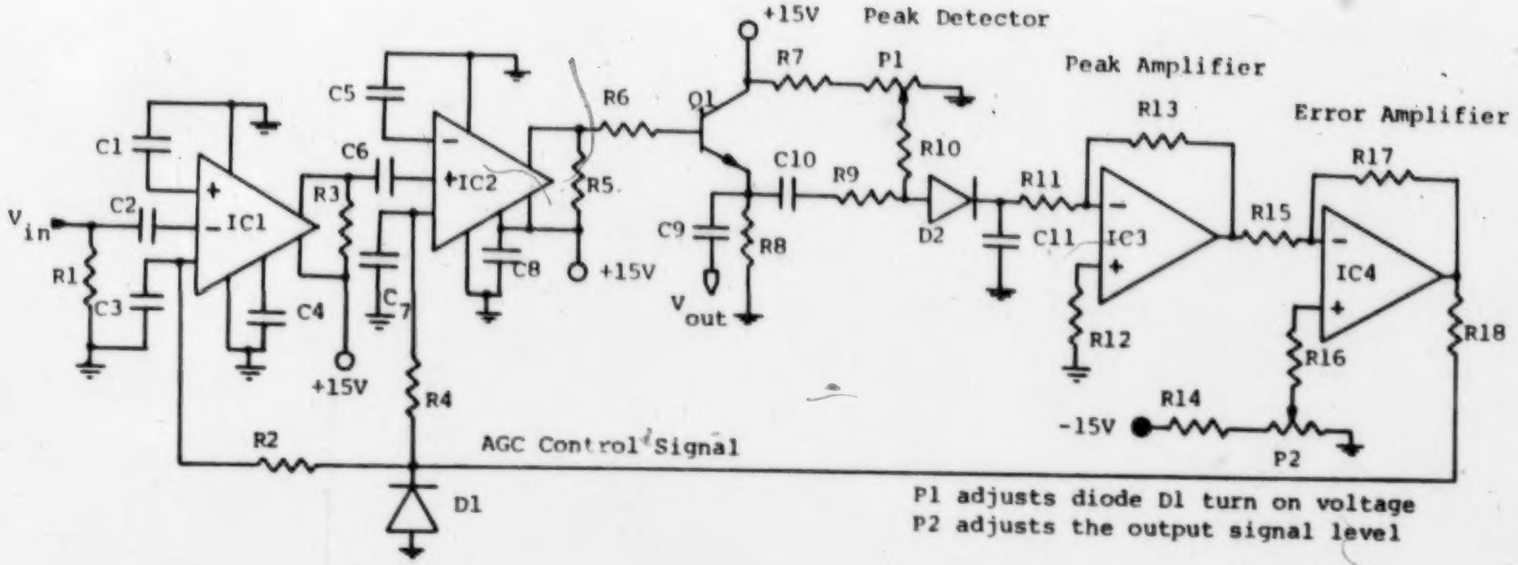
The preamplifier is followed by a main signal amplifier (as shown in Figure 4.11) which has 50 ohm input and output impedances, a large signal bandwidth, and automatic gain control (AGC). AGC is necessary as the average optical input power to the APD can vary, possibly over a range of 20 dB. The output of the main amplifier should vary as little as possible with varying input signal strength, since proper operation of the following decoding stage is critically dependent on its input signal levels. Noise is no longer a major consideration as it was for the preamplifier, therefore the bandwidth of the circuit can be large. Specifically, it should be larger than the channel and preamplifier bandwidth in order to minimize signal distortion. The AGC amplifier was designed and constructed by C. Kelly [44]. It has a maximum output signal level of 0.5 V peak to peak, and a bandwidth of approximately 45 MHz. It was therefore possible to use this amplifier without any modifications.

4.5 Decoder

Figure 4.12 outlines the overall decoding and frame finding network for the 16B9Q decoder. For the purposes of this project, it was not necessary to implement the frame finding circuitry and therefore only the decoding portion, shown in Figure 4.13 was constructed (frame synchronization was achieved by a trial and error process). For completeness, however, a description of the operation of the entire decoder is given below. Table 4.4 lists the components used in the decoder circuits.

The decoder clock generator, similar to the coder clock generator, is shown in Figure 4.14. The input clock signal is buffered by line receiver, IC1. IC3 provides divided by two, four, eight, and sixteen signals, which are retimed by IC4. IC2 is a debounced switch connected to the reset pin of the binary counter, IC3, which can be used to reset the counter, to facilitate the trial and error process of finding frame. The remainder of the circuit, ICs 5 through 8, function in an identical manner to the corresponding chips in the coder clock, to produce the 9/16 clock signal.

Figure 4.15 illustrates the four level to two level decoder. Signals at the positive and negative input terminals of A/D comparators IC9 and IC10 are compared, resulting in a high level at the output when the voltage at the positive terminal is greater than that at the negative terminal. The four level input signal is compared to reference voltage levels set by potentiometers P1, P2, P3, and 3.3 V zener diodes. IC11 converts the resulting output signals



P1 adjusts diode D1 turn on voltage
P2 adjusts the output signal level

R1 = 50Ω
R2,14 = 4.7 kΩ
R3,5 = 100Ω
R4,7 = 10 kΩ
R6 = 470Ω
R8,18 = 1.0 kΩ
R9 = 2.7 kΩ
R10,15,16 = 20 kΩ

R11,12 = 39 kΩ
R13 = 390 kΩ
R17 = 270 kΩ
P1 = 1.0 kΩ
P2 = 5.0 kΩ
C1,5,10,11 = 0.1 µF
C2 = 1.0 µF
C3,4,7,8 = 22 µF/16V

C6 = 2.2 µF
0.1 µF
C9 = 470 µF/16V
1 µF
D1,2 = 1N4148
Q1 = 2N5943
IC1,2 = MC1590
IC3,4 = MC741

Figure 4.11. The experimental main amplifier, peak detector, and automatic gain control circuits.

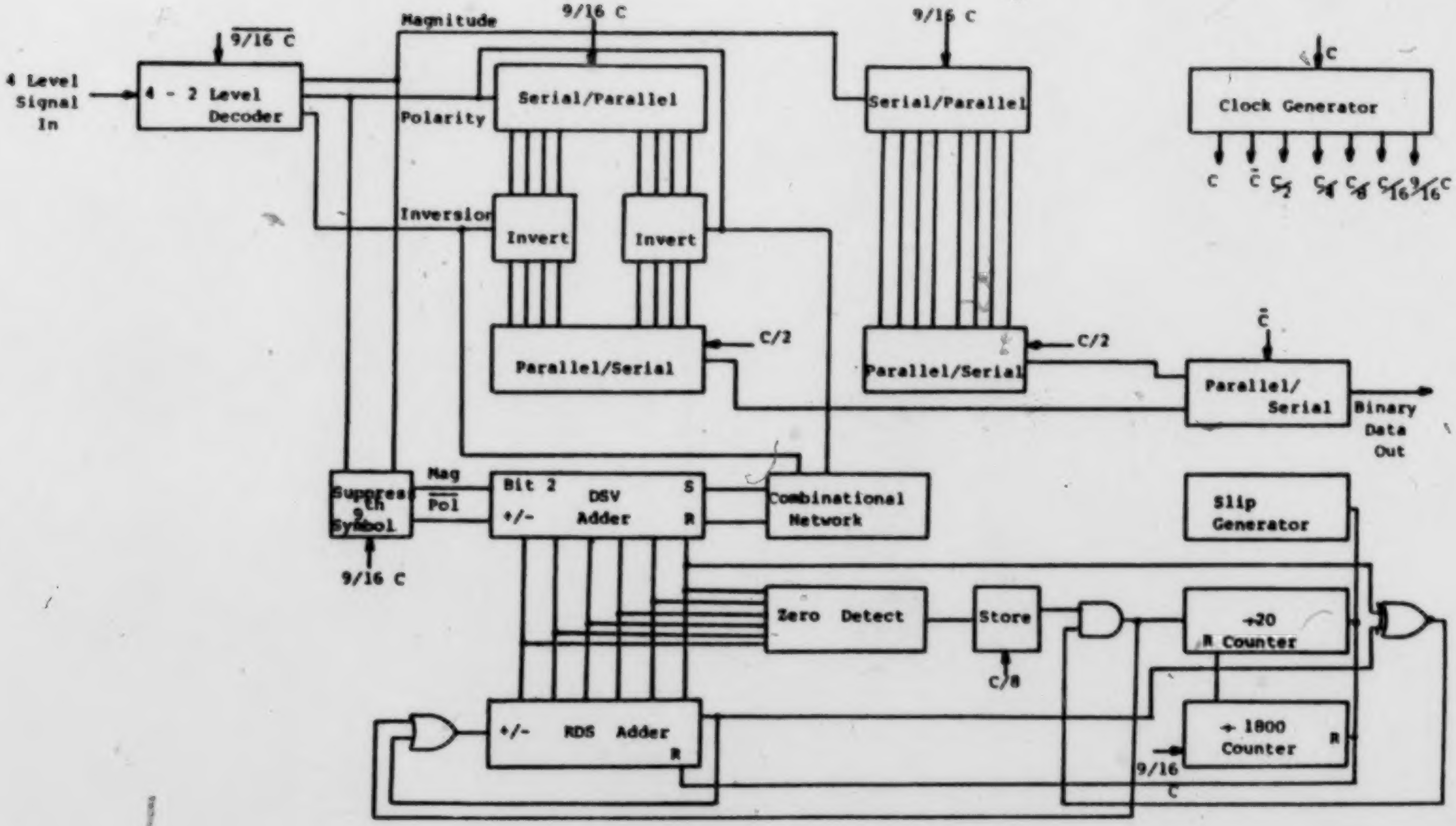


Figure 4.12. Block diagram of the overall 16B9Q decoder.

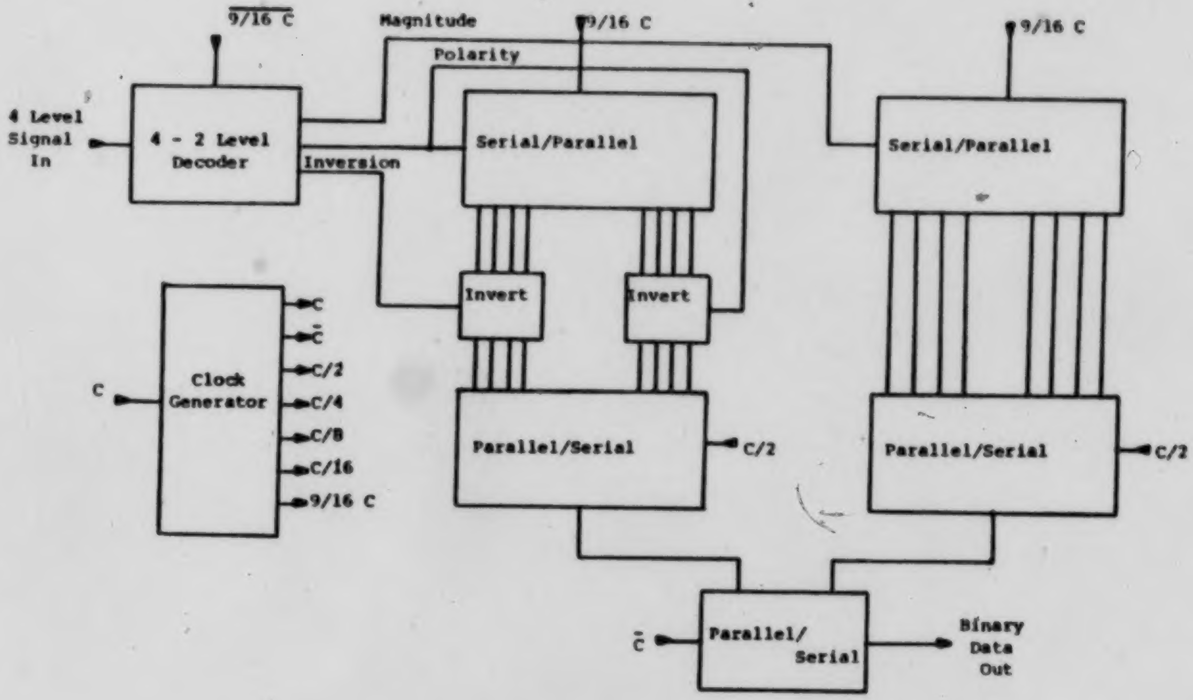


Figure 4.13. Block diagram of the implemented 16B9Q decoder.

Table 4.4. List of Decoder Components.

Component Number	Component
IC 11	MC10104
IC 13	MC10107
IC 17	MC10109
IC 15,19	MC10113
IC 1	MC10116
IC 8	MC10124
IC 5	MC10125
IC 21	MC10131
IC 14,16,18,20,22-26	MC10H141
IC 3	MC10154
IC 12	MC10165
IC 9,10	MC1650
IC 7	74F161
IC 6	NE564
R 1,2	10 k Ω
R 3,4	1 k Ω
R 5	9.1 k Ω
R 6	2.7 k Ω
R 7	100 Ω
R 8	560 Ω
C 1,3,4	0.01 μF
C 2,5	0.1 μF
C 6,7	820 pF
C 8	20 pF varicap
P 1,2,3	10 k Ω potentiometers

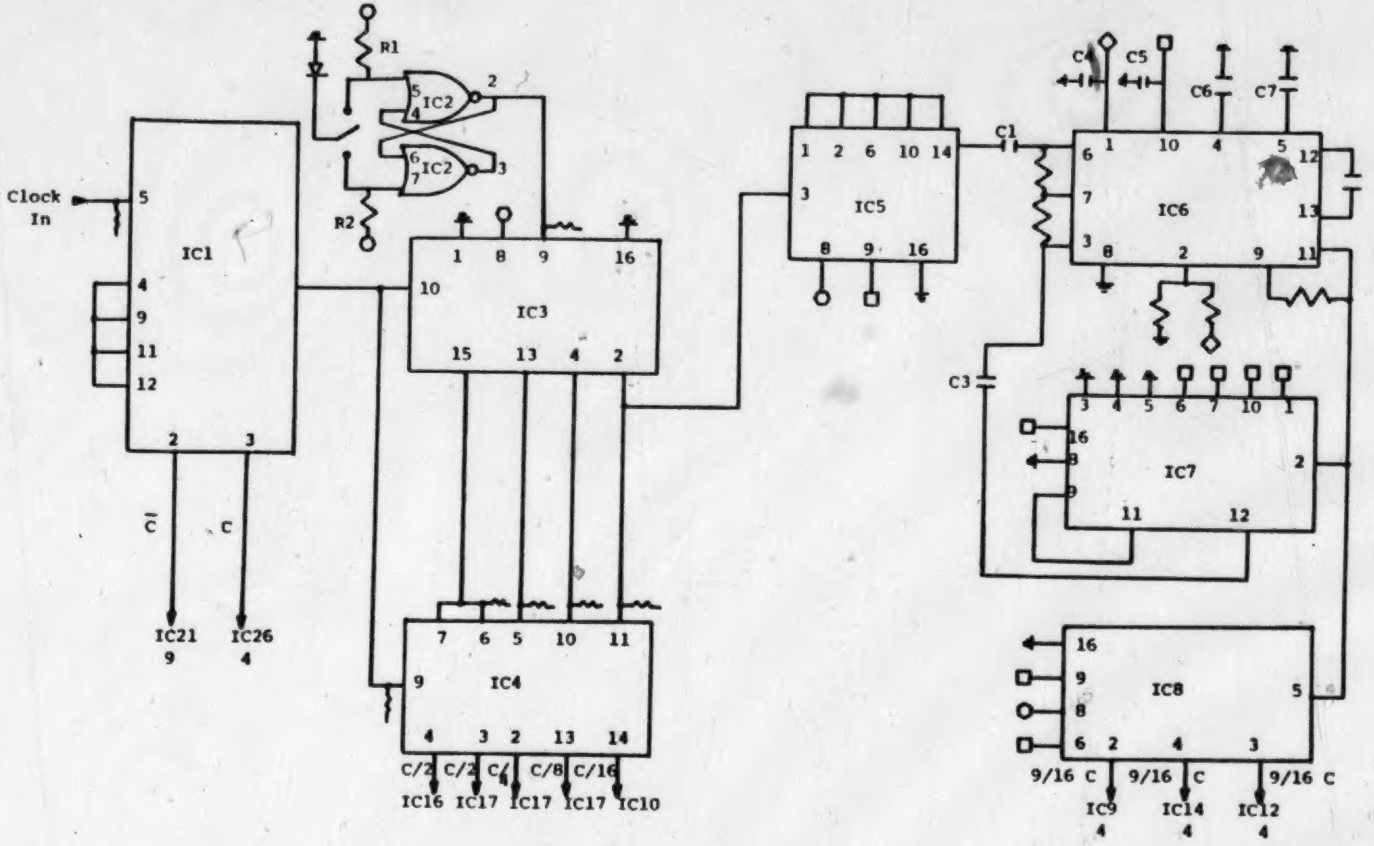


Figure 4.14. The decoder clock generating circuit.

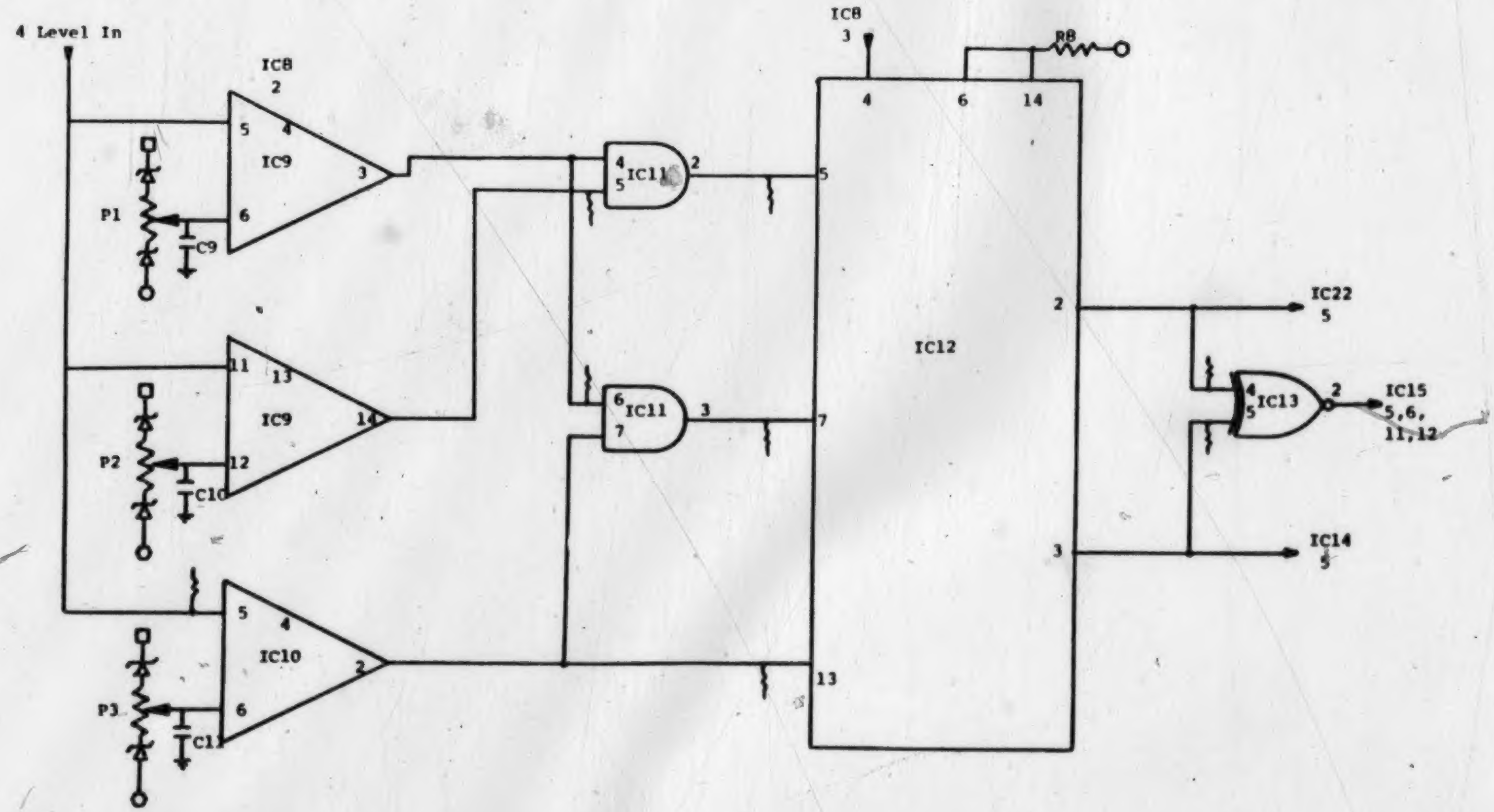


Figure 4.15. The four level to binary converter circuit.

to a pattern that when presented to priority encoder IC12, will cause the two output bit streams of IC12 to be identical to the input data of the binary to quaternary converter (Figure 4.8). IC13 then provides the signal which will be used to determine how to decode (that is, which bits to invert) the 16B9Q encoded binary signal.

The removal of the 16B9Q code is performed by the circuit shown in Figure 4.16. IC17 and IC21 create the control signals for shift registers ICs 14,16,18,20,22,23,24,25, and 26. The magnitude and polarity bit streams recovered by the A/D comparators and priority encoder, are serially shifted into ICs 22,24 and 14,18 , respectively. Once the entire frame has been clocked in (eight bits of magnitude and eight of polarity), shifting halts while the ninth symbol is applied to inverters IC15 and IC19. The polarity bits are thus restored to their original form. The entire frame is then loaded into ICs 16,20,23, and 25 and shifted into IC26 which multiplexes the magnitude and polarity bits into a serial output stream. This signal is identical to the input data to the coder.

Correct recovery of the binary signal depends upon synchronization of the ninth symbol with C/16, which can be implemented using the RDS and DSV of the incoming data. If the signal is properly framed, the current DSV and the previous RDS will be of opposite polarity. If, however, framing is incorrect, violations of this rule will occur frequently. After a certain number of errors have been identified the slip generator can be inhibited, causing the incoming signal to slip by one symbol. This process should continue until correct frame has been found and no further errors are generated. This method relies upon the knowledge of the correct value of RDS, therefore, a mechanism for finding this value is incorporated into the synchronization strategy.

Referring to Figure 4.12, incoming data to the decoder is summed in the DSV adder, with the value of the ninth symbol being added to the DSV in much the same manner as in the coder. The combinational network processes the first word and second word inversion signals to extract the portion of the ninth symbol belonging to that word. That is, the adder is initialized to a ±2 at the beginning of the first word and to a ±1 at the beginning of the second. The polarity of the initialized value depends upon the actual word being processed. For example, if the first word is being processed and it was inverted by the coder prior to transmission, the adder in the decoder will be initialized to a +2. If this word was not inverted, the initializing value will be -2. Thus, the weighting of the ninth symbol on each word is exactly the same as in

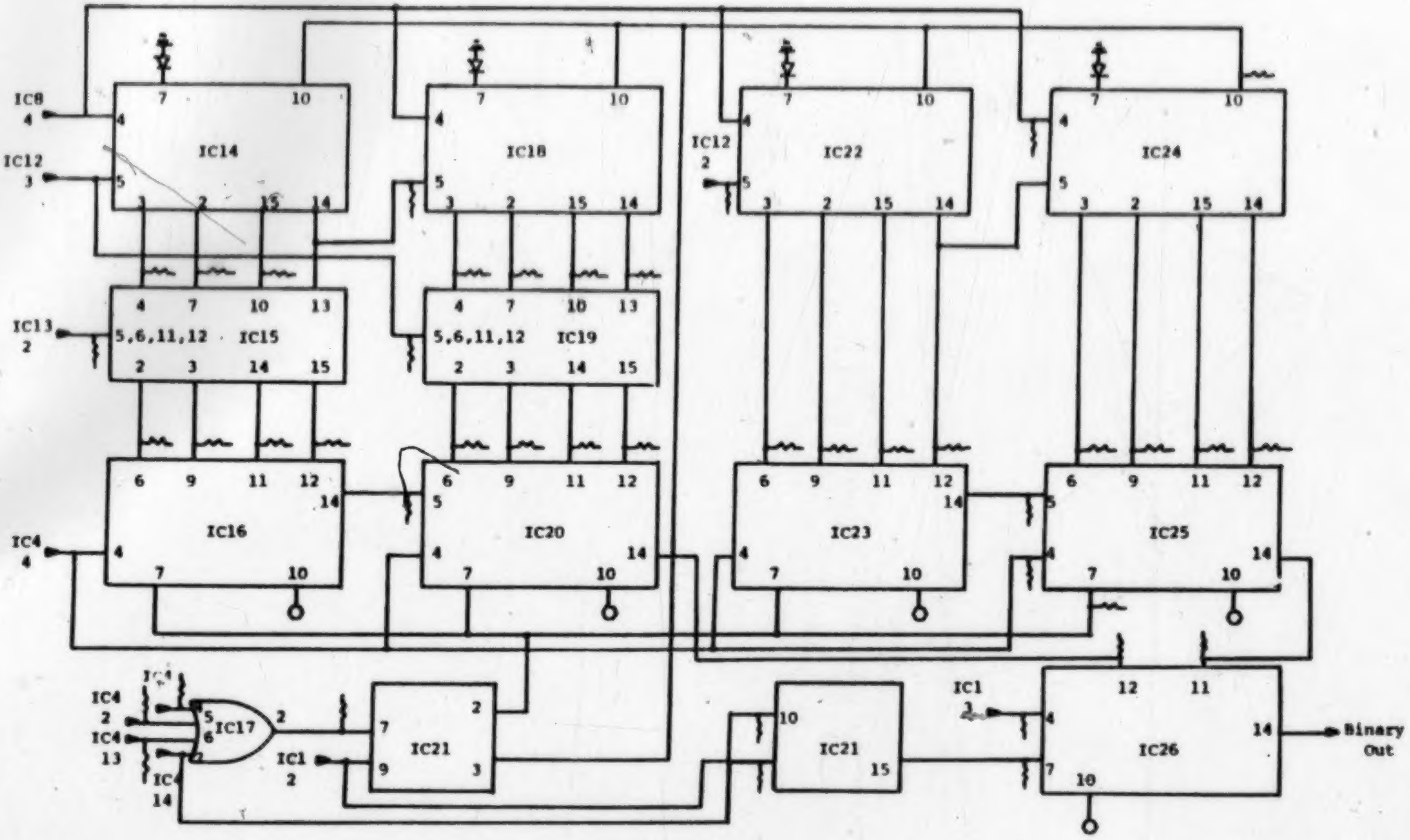


Figure 4.16. The decoder inversion and output circuit.

suppressed at the adder inputs. The DSV is added to the RDS to determine the current value of RDS. An initial RDS value of zero is assumed. Since a DSV of zero is assumed to be positive, but should not generate an error if the RDS is positive, a zero detection circuit is used to inhibit the input to the error counting circuit and to the RDS correction circuit when a DSV of zero is encountered. The polarities of the DSV and RDS are compared by an exclusive NOR. If they are the same, the ÷20 counter which keeps track of polarity violations, increments by one and the RDS value is adjusted towards zero by one (if it is negative a 1 is added; if it is positive a 1 is subtracted). This process is repeated until twenty errors have been counted. At this point the ÷20 counter generates an output which inhibits the input to the slip generator for one symbol period, resets the RDS adder and the ÷1800 counter to zero, and begins the entire process again. This continues until frame has been found and no further errors are generated. It is possible that line errors will occur that will cause the DSV and RDS to be of the same polarity and thus generate an error, even though the decoder is fully synchronized. These errors are accumulated in the ÷20 counter and will eventually cause a slip to occur. In order to prevent such spurious reframe, after 1800 symbols have been decoded, the ÷20 counter is reset. In the case of true misalignment, errors will occur much more often than this and the slip mechanism will be able to operate properly.

CHAPTER 5

EXPERIMENTAL RESULTS

Following the design and construction of the 16B9Q coder and decoder circuits, an experimental system was set up as shown in Figure 5.1. The system was used to verify code properties and to establish the usefulness of the 16B9Q line code. This chapter discusses the experimental results, and compares them, where appropriate, with theoretical ones.

5.1 The Experimental System Topology

An HP 3762A Data Generator was used to provide both the input data to the coder, as well as the 44.736 MHz system clock (DS-3 rate). The data generator has two clock outputs: an ECL level output, used as the coder clock source, and a 50 ohm clock output used at the receive side of the system as the decoder clock. A suitable delay in the 50 ohm clock line is provided by an HP 1910A Delay Generator, which is adjustable from 0 to 100 ns in increments of 5 ns. An HP 1906A Rate Generator amplifies and level translates the incoming clock signal so that it can properly trigger the delay generator. Since the output of the delay generator is a TTL signal, it requires level shifting and buffering by an HP 3763A Error Detector before it can be used in the decoder. The data generator produces a binary signal in the form of 16 bit words (the particular word chosen is set with switches on the instrument's front panel) as well as three pseudo random bit sequences (10^{23}-1, 10^{15}-1, 10^{10}-1). The data signal enters the coder, is processed according to the 16B9Q coding rules, and is transformed into a 25.164 Mb/s four level output signal which modulates the laser transmitter. Light pulses are transmitted through 5 km of multimode optical fiber to the APD detector, which is followed by a transimpedance preamplifier providing the initial received signal amplification, and by an AGC amplifier providing constant output signal levels. The decoder accepts the recovered four level signal and the delayed clock, and sends the decoded binary bit stream to the error detector, which determines and displays a bit error rate (BER) reading corresponding to the difference between the original and decoded data signals. To save circuit design and construction time, framing is carried out by adjusting the delay generator until synchronism is obtained. A problem exists with this method in that, due to the PLL in the clock generating circuitry of the decoder, adjustment of the delay generator does not result in a proportional delay in the 9/16 clock signal. The PLL maintains the phase difference between the 9/16 clock signal and the C/16

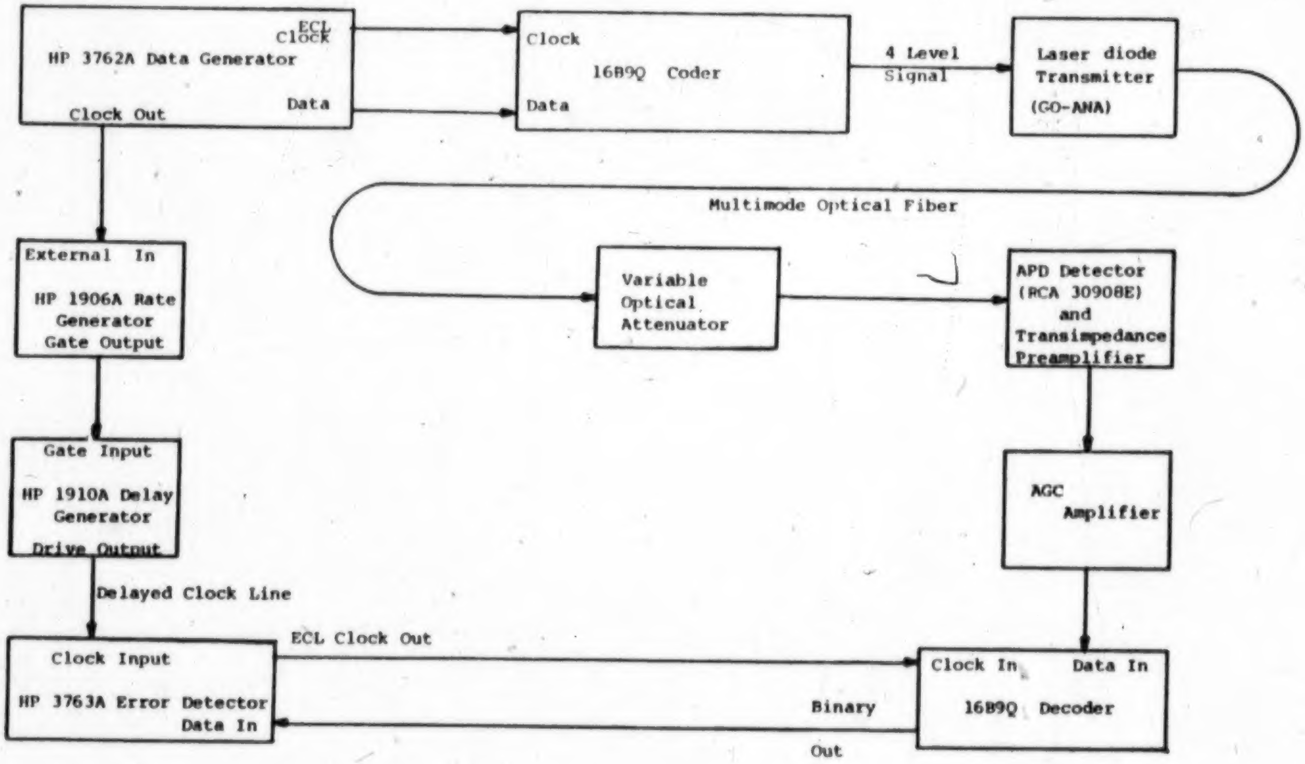


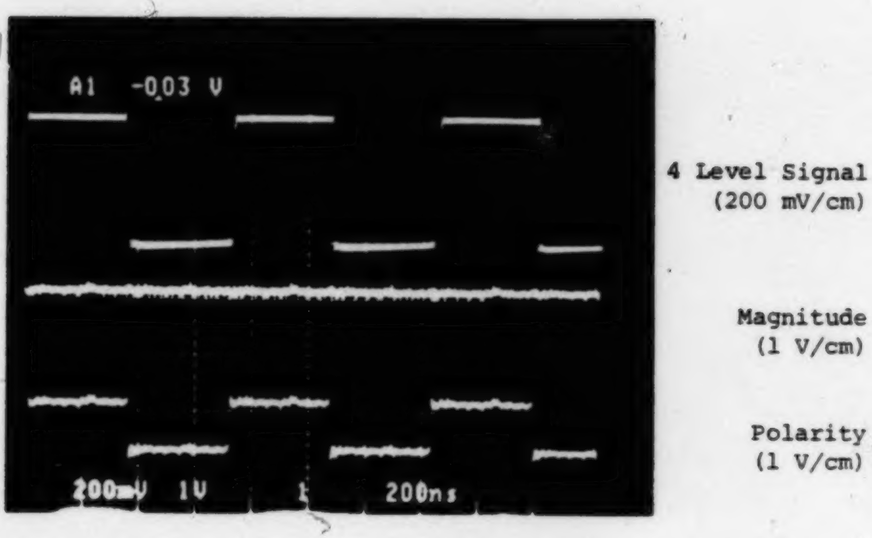
Figure 5.1. The experimental 16B9Q system test set up.

signal (which is the delayed signal) close to zero. Thus, the 9/16 clock waveform may shift by entire clock cycles, making it difficult to obtain a small delay increment. If the ninth symbol is not detected in exactly the right place, synchronism is not attained. Therefore, it is very difficult to synchronize the decoder.

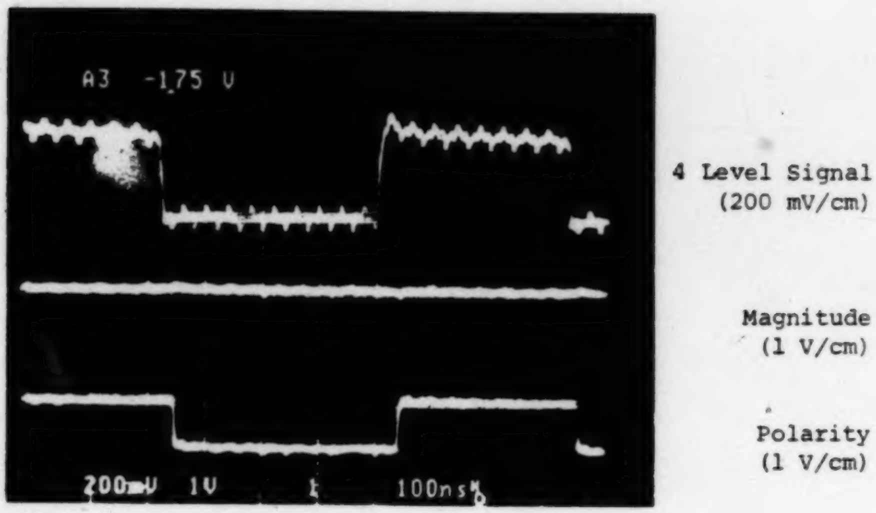
5.2 Coder and Decoder Operation

Figures 5.2 to 5.4 demonstrate the operation of the 16B9Q coder and decoder. Upon accepting an input data signal, the coder separates it into two parallel bit streams corresponding to the magnitude and the polarity of the output signal. These bit streams are processed and then combined into one four level signal, which is transmitted over 5 kilometers of multimode optical fiber. Following detection and amplification, the decoder separates the four level signal into its constituent magnitude and polarity bit streams which are further processed to reproduce the original data signal. Figures 5.2 to 5.4 display the coder and decoder waveforms for three different 16 bit input words. It is evident from these that the system functions properly. The coder and decoder signals are identical, with the exception of noise spikes on the four level signals at the decoder. These spikes are caused by 9/16 clock frequency pick up by the decoder circuits, due to insufficient circuit shielding and power supply decoupling. For the most part they did not cause any decoding problems. BER readings of 10^{-9} were obtained for each of the input words, indicating satisfactory system performance.

Figure 5.5 shows the coder and decoder waveforms for an input 2^{23}-1 PRBS. The coder output is a four level pseudo random symbol stream with a perfectly open and symmetrical eye pattern. The output of the preamplifier, however, is degraded due to inadequate preamplifier frequency response. Although the two lower eyes are open, the top one has a very slow rise time resulting in a partially closed eye. This problem is caused by insufficient preamplifier bandwidth rather than fiber dispersion, since the fiber has a bandwidth of approximately 0.441/(1.5 ns/km x 5 km) = 58 MHz. As a result of the slow rise time, level transitions occur over an interval of time rather than at one point, causing timing jitter. It is thus easier for the comparators of the decoder to make an incorrect decision, and consequently the error rate increases. In practice, it proved to be impossible to obtain a BER reading for this case due to the combination of this problem and the synchronization difficulty.



(a)

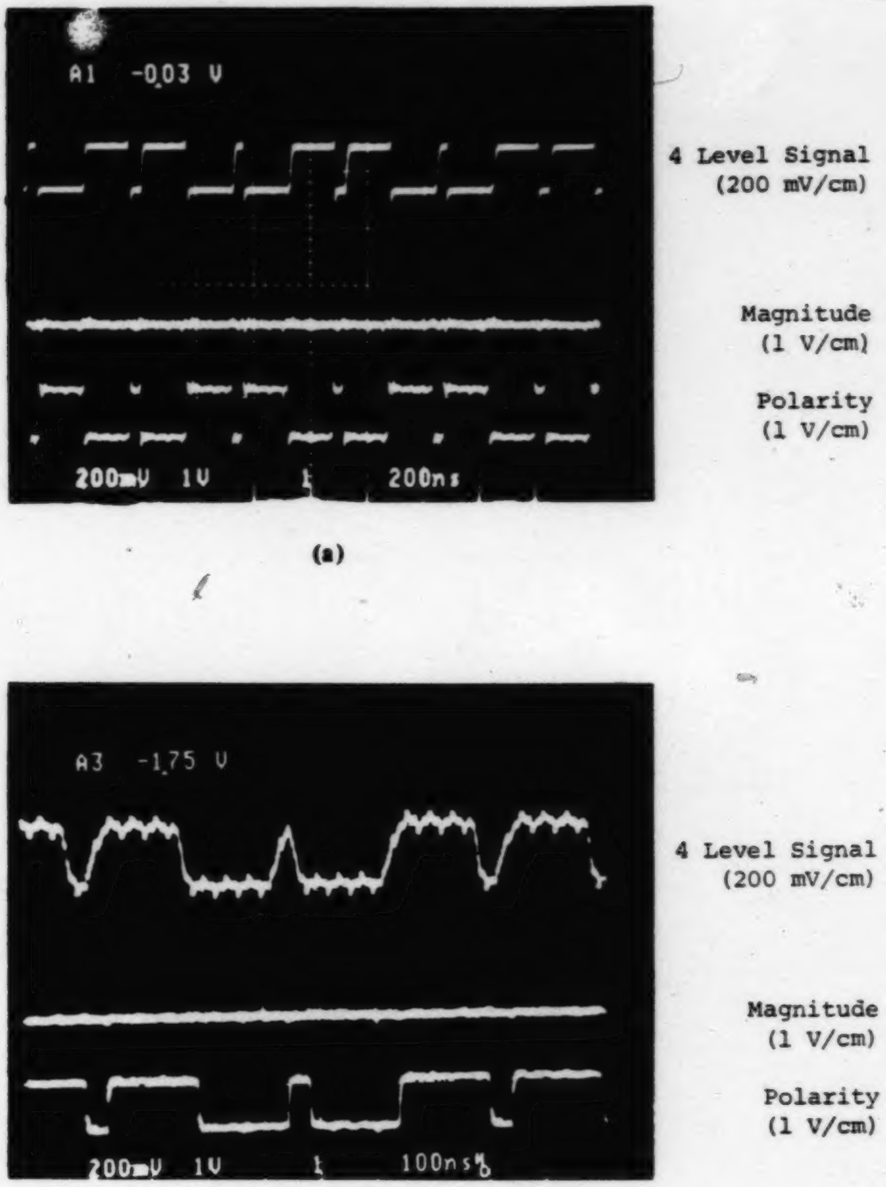


(b)

Figure 5.2. The four level 16B9Q coded signal and its constituent magnitude and polarity waveforms for an input data word of 1111111111111111.

(a) Coder waveforms.

(b) Decoder waveforms.



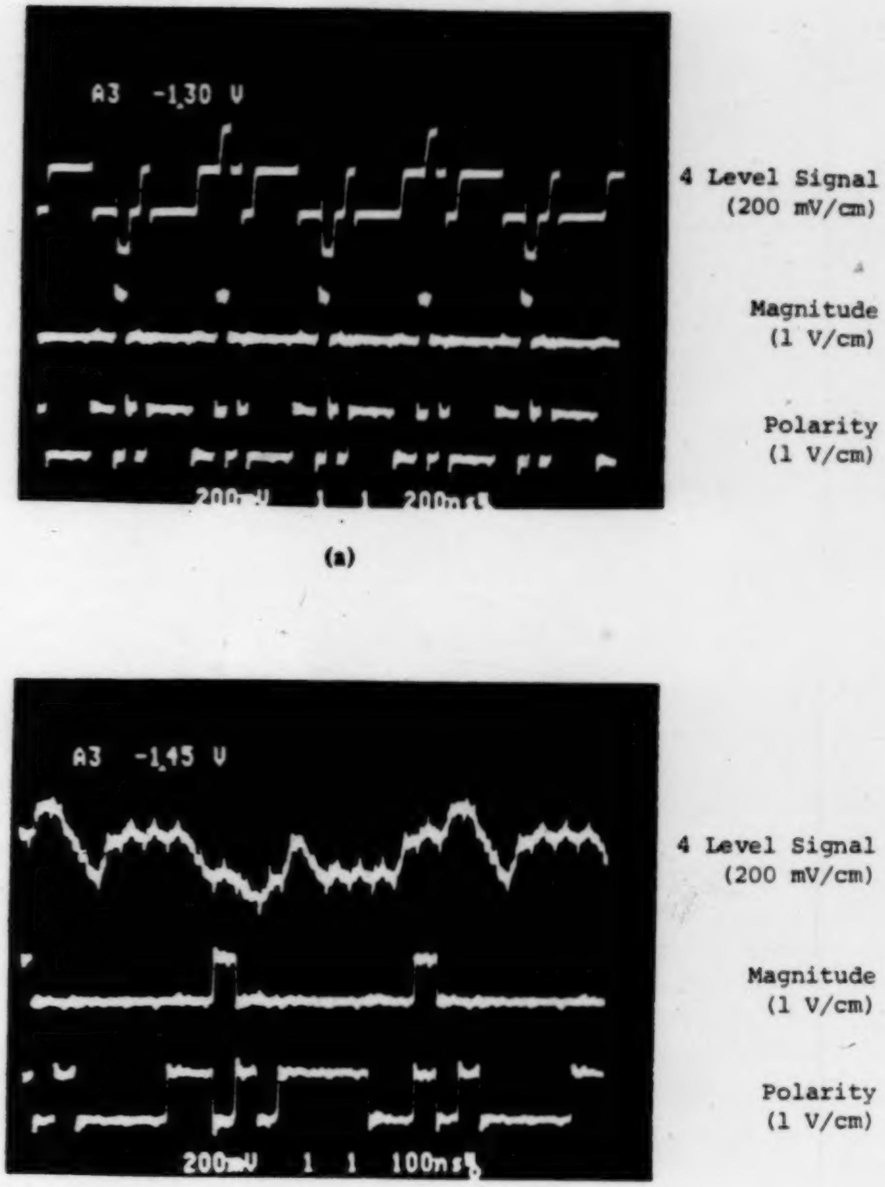
(a)

(b)

Figure 5.3. The four level 16B9Q coded signal and its constituent magnitude and polarity waveforms for an input data word of 0000000000000000.

(a) Coder waveforms.

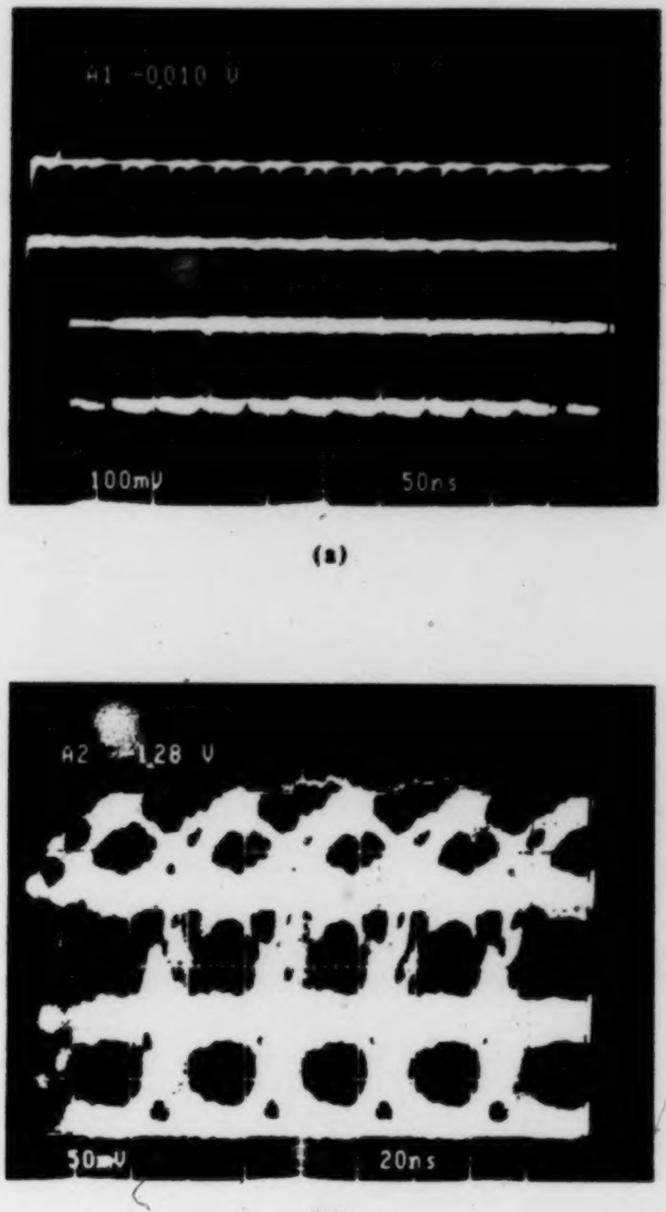
(b) Decoder waveforms.



(a)

(b)

Figure 5.4. The four level 16B9Q coded signal and its constituent magnitude and polarity waveforms for an input data word of 0001100000000000.
(a) Coder waveforms.
(b) Decoder waveforms.



(a)

(b)

Figure 5.5. The four level 16B9Q coded signal for an input 2^{23}-1 PRBS.

(a) Coder waveform.

(b) Decoder waveform.

5.3 Operation over Dispersion Limited Fiber

The advantage of a multilevel line code in a dispersion limited system is its ability to transmit a larger amount of information (i.e., a higher bit rate) than is possible with a binary line code.

Figure 5.6 displays theoretical bit rate versus range curves for a highly dispersive multimode fiber (Siecor Super-Fat Fiber Cable #155 - 5 MHz-km bandwidth, 35 dB/km attenuation) coupled to an ILD source (-3 dBm coupled optical power) and a short wavelength APD detector. These curves were obtained using the method described in Appendix B. It is evident that a binary coded signal can be transmitted at a bit rate of 44.736 Mb/s over the test fiber for only 110 m before it becomes dispersion limited (i.e., significant ISI occurs). A signal encoded with the 16B9Q line code, however, has a range of 200 m under the same conditions. This was verified experimentally by transmitting both the binary and 16B9Q signals over 200 m of the Siecor fiber.

Figures 5.7 and 5.8 show the results of this test. Figure 5.7 displays the waveforms of the binary system for both an arbitrary 16 bit input word and a 2^{23}-1 PRBS. Figure 5.8 shows the corresponding waveforms of the 16B9Q system. From Figure 5.7 it is evident that a large amount of ISI occurs in the binary system. The high frequency pulses in part (a) of the figure do not reach either their maximum or minimum values. The build up of ISI can be observed in the short pulse which follows the longer string of ones. The minimum level of this pulse is higher than the minima of the short pulses preceding the string of ones. Thus, decoding errors are more likely to occur, resulting in degraded system performance. The receiver eye pattern of the pseudo random bit sequence in part (b) resembles a multilevel pattern more than a binary one. Dispersion has spread the pulses to a great extent, causing a slow rise time, closing the eyes, and increasing the error probability. In Figure 5.8, however, for both the 16 bit word and the pseudo random sequence, the received waveforms are practically identical to the transmitted ones, with very little degradation occurring. The pseudo random eye pattern has open eyes and well defined level transitions. In this case, satisfactory system performance can be obtained.

Thus, a multilevel line code, in particular the 16B9Q line code, can be used to increase the capacity of dispersion limited optical fiber systems.

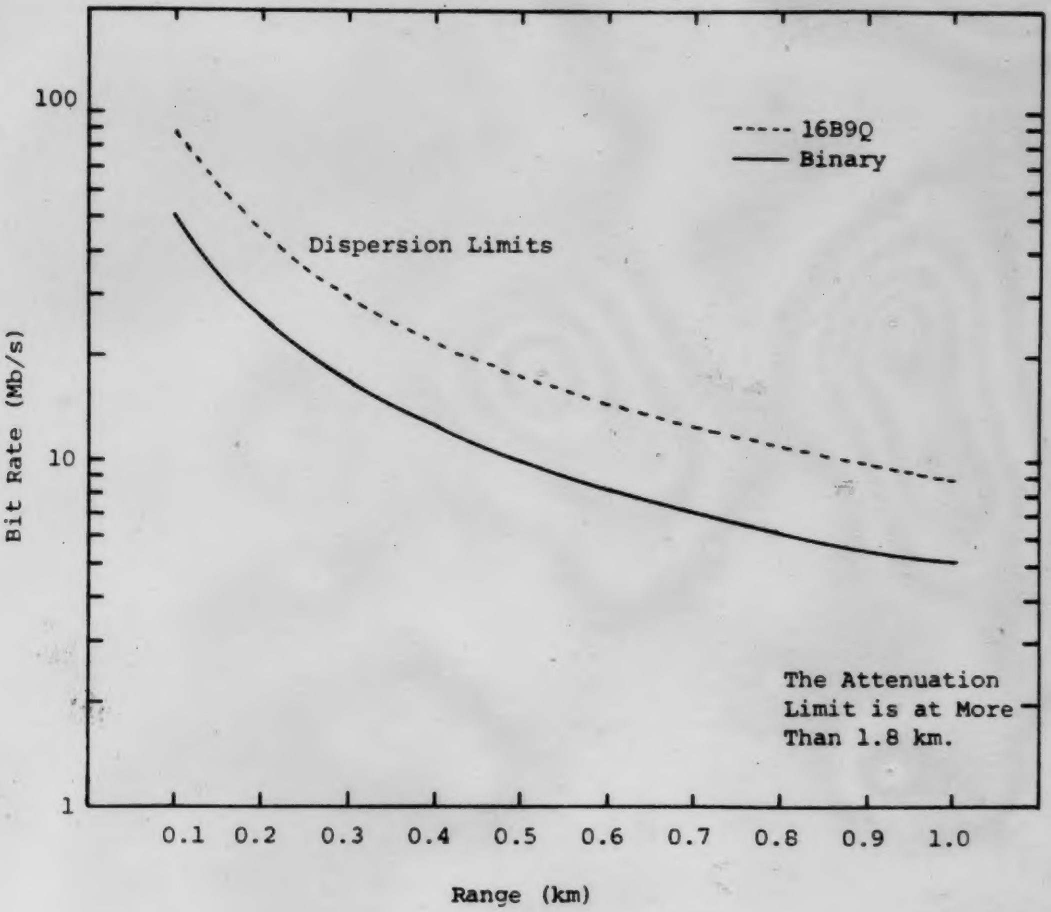
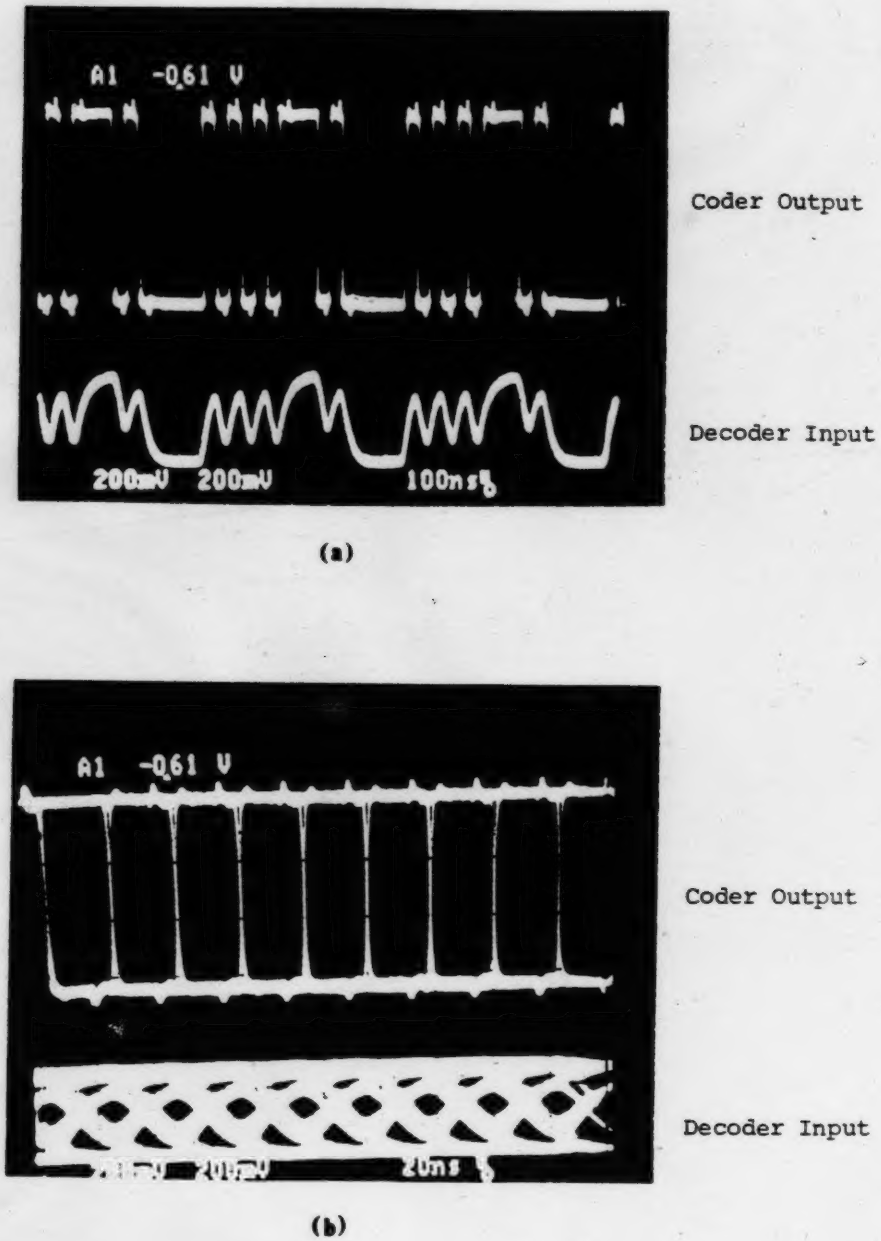


Figure 5.6. Theoretical bit rate versus range curves for the experimental 16B9Q system utilizing highly dispersive multimode fiber.



(a)

(b)

Figure 5.7. Experimental results of transmitting a binary encoded signal at a bit rate of 44.736 Mb/s over a 25 MHz bandwidth multimode fiber.

(a) Coder input is a 16 bit word.

(b) Coder input is a 2^{23}-1 PRBS.

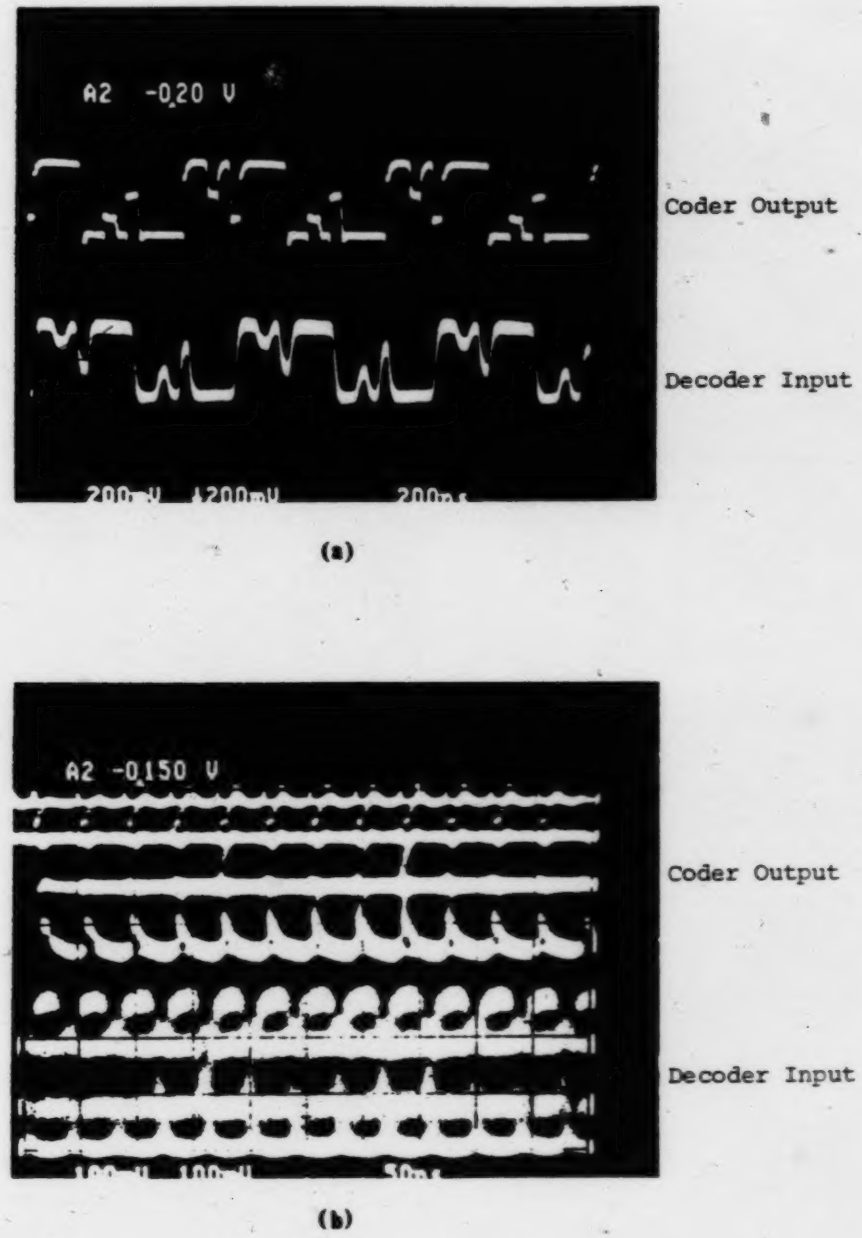


Figure 5.8. Experimental results of transmitting a 16B9Q signal at a bit rate of 44.736 Mb/s over a 25 MHz bandwidth multimode fiber.

(a) Coder input is a 16 bit word.

(b) Coder input is a 2^{23}-1 PRBS.

5.4 Power Spectrum

Figures 5.9 to 5.11 show the power spectra of 16B9Q encoded pseudo random sequences. Figures 5.9 (a), (b) and (c) display the spectra of the coder output signals for the three different PRBS lengths, 2^{23}-1, 2^{15}-1, and 2^{10}-1, respectively. It is evident that all three are essentially identical, proving that given any input sequence, the 16B9Q encoding process produces a signal with the same power distribution. The initial spike in these photographs is produced by the spectrum analyzer and indicates the zero frequency point. The lines seen at 25.164 MHz and its higher harmonics have been found to be a result of clock frequency pick up by the coding circuit (i.e., not enough shielding) and are not a feature of the code. The power level drops as the frequency increases, to -45 dBm at 25.164 MHz, and the sidelobes are increasingly attenuated. Figure 5.10 shows the low frequency end of the spectrum. Due to the resolution of the spectrum analyzer used (HP 8557A - it only measures down to 10 kHz) and its 'zero spike', it is not possible to see the dc value of the power spectrum. However, the lowest point occurs at a power level of -90 dBm; therefore, it can be assumed that the power at dc is essentially zero. The power level at the symbol frequency is also quite low, and the slope of the curves on either side of it is not too steep, enabling the main lobe to be filtered out easily. Thus, the preamplifier bandwidth can be limited to the frequency range of the main lobe. Figure 5.11, the output of the preamplifier for one of the pseudo random sequences following transmission over 5 km of the Northern Telecom multimode fiber, shows the effect of such filtering. Due to the small bandwidth of the preamplifier all the second and higher side lobes are eliminated, while the first side lobe is greatly attenuated. From this figure, it is also evident that the frequency response of the experimental preamplifier is not flat as it should be ideally. The main lobe has a slightly different shape than in the unfiltered case (Figure 5.9), due to the greater attenuation of the higher frequencies.

Figure 5.12 shows a numerically generated power spectrum [38] for the 16B9Q line code. It was obtained by generating a random binary bit stream (with the probability of 0s and 1s being 50%), coding it according to the 16B9Q line coding rules, estimating the sample autocorrelation function, and using this in Bennett's formula [38]

$$W(f) = R(0) + 2 \sum_{j=1}^{N} R(j) \cos(2\pi j f) \tag{5.1}$$

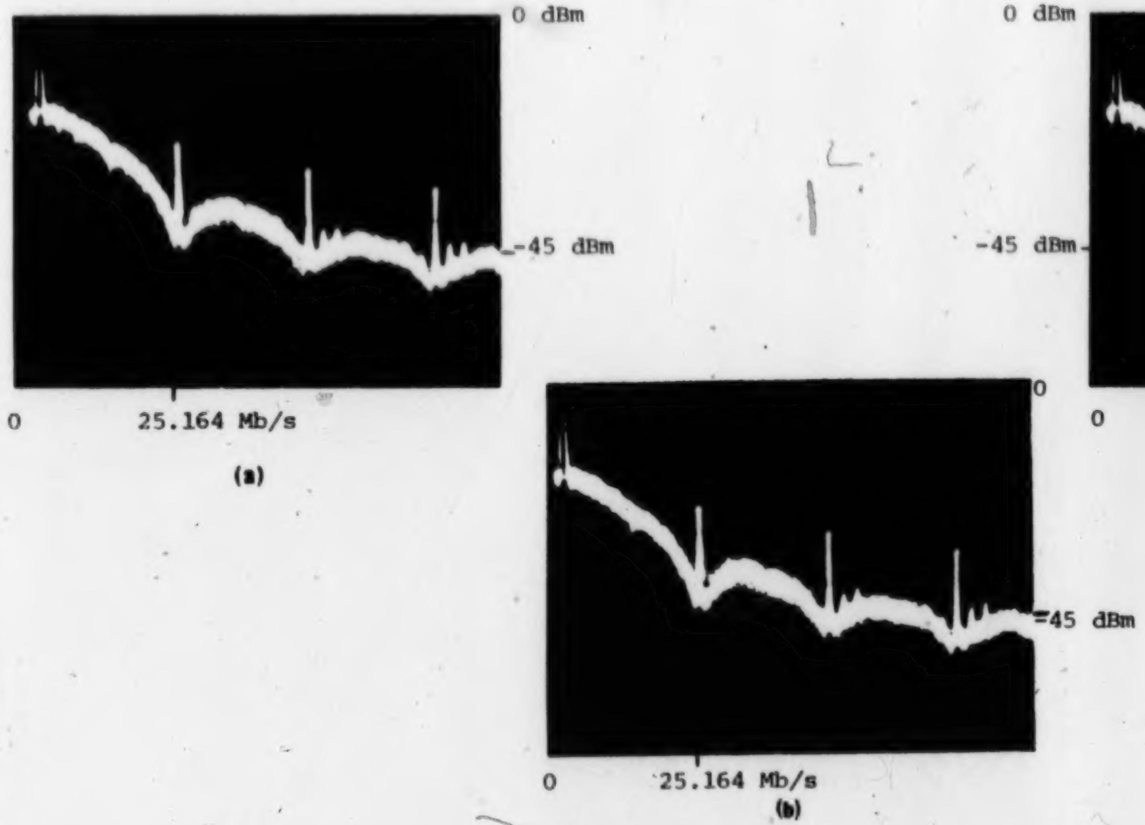


Figure 5.9. Power spectra of three different 16B9Q encoded PRBS signals at the coder output.

(a) 2^{11}-1 PRBS. (b) 2^{15}-1 PRBS. (c) 2^{10}-1 PRBS.

The horizontal scale is 13.33 MHz/cm and the vertical scale is 13.33 dB/cm.

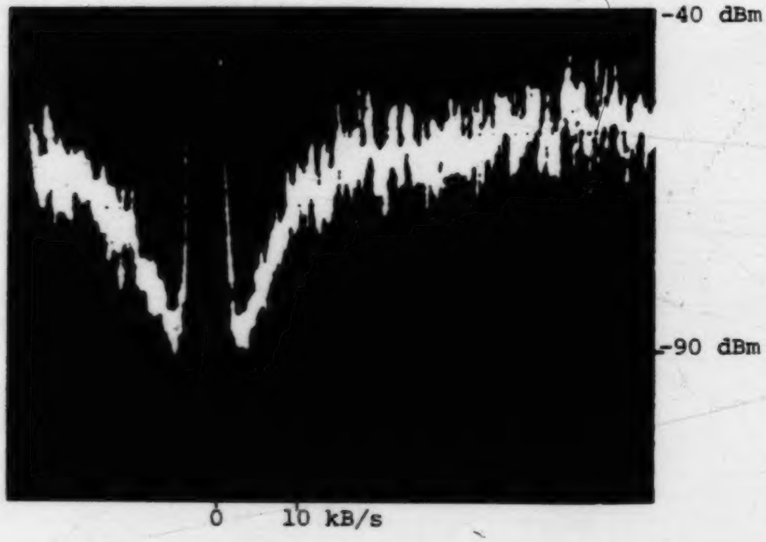


Figure 5.10. **Low frequency spectrum of a 16B9Q encoded 2^{23}-1 PRBS.**
The horizontal scale is 10 kHz/cm and the vertical scale is 10 dB/cm.

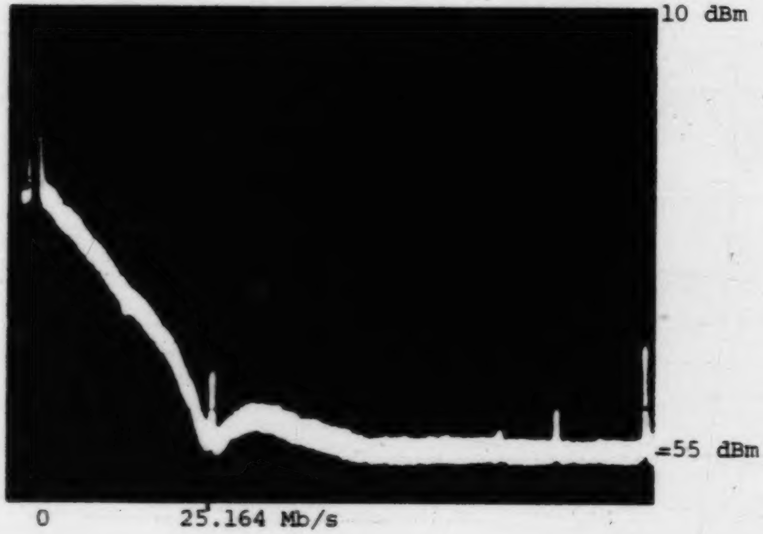


Figure 5.11. **Power spectrum of a 16B9Q encoded 2^{23}-1 PRBS at the output of the preamplifier.**
The horizontal scale is 10 MHz/cm and the vertical scale is 10 dB/cm.

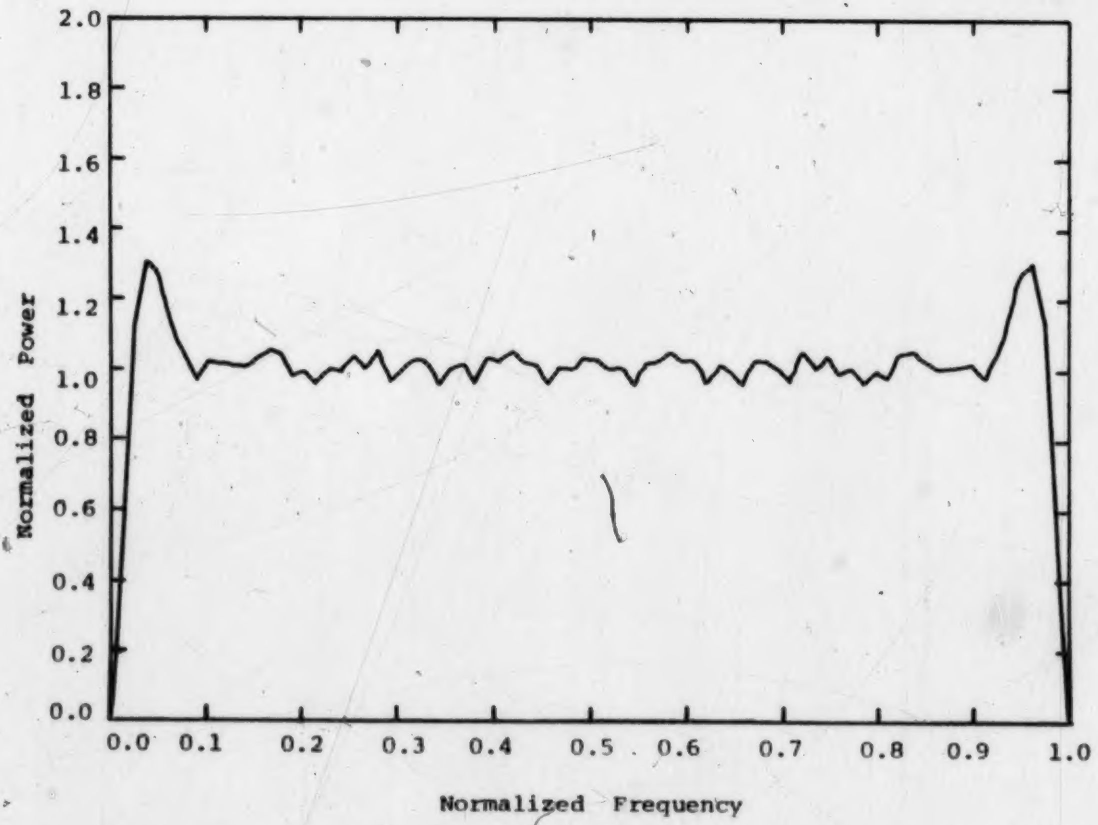


Figure 5.12. Computer simulated power spectrum of the 16B9Q line code [38].

to determine the sample spectrum.

A comparison of the power spectra of the 16B9Q line code derived by three different methods (analytical expression of Chapter 3 multiplied by sinc^2(f), output of the actual coder, and the computer generated estimate of Figure 5.12 also multiplied by sinc^2(f)) shows that they are essentially identical. Nulls are found at dc and at harmonics of the symbol frequency, and the spectrum is fairly flat over the entire range. The only discrepancy is that the spectrum of the analytical expression of Chapter 3 goes to a minimum but not to zero at dc and at the symbol frequency. The entire power spectral density curve is raised on a pedestal of magnitude equal to the average power of the signal. From the computer generated curve and the output of the coder, it is known that in fact the minimum should be zero; thus it is possible that the analytical method is not entirely accurate. Due to the complexity of the calculations involved, however, it would be difficult and time consuming to determine where the analytical method is in error. Further work in this area is thus required.

5.5 Bit Sequence Independence

As mentioned in Chapter 1, a bit sequence independent code will transform any source data into a line signal that can be transmitted over the channel without performance degradation. The statistics of such a coded signal will therefore be the same for any input data, giving rise to an identical distribution of power in the frequency domain. This can be observed from Figure 5.9. The input data for each photograph is a different pseudo random binary sequence and each results in an identical power spectrum at the output of the coder. Thus, the 16B9Q line code is bit sequence independent.

5.6 Timing Content

To most efficiently transmit a signal over a communication channel, it is essential that the receiver be able to recover a timing signal from the data itself. This process is greatly simplified if discrete spectral lines exist in the power spectrum. Even if they are not present, as in the case of the 16B9Q line code, timing extraction can still be accomplished. The presence of level transitions suggests that timing information is present in the signal, despite the absence of spectral lines. This is due to the fact that the continuous spectral component is not a random signal, but has hidden phase relationships between its components in different frequency bands.

These relationships are described by the cyclostationarity of the coded process. If the signal is subjected to non linear filtering (this transforms all transitions into positive ones) a discrete spectral line emerges at the signalling frequency. The application of this waveform to a BPF or PLL allows the clock signal to be regenerated. As a result of the 16B9Q coding process, a sufficient number of level transitions occur to ensure good clock recovery statistics. In the worst case situation, eight identical symbols can be transmitted before a level transition occurs. The probability of this situation is determined from the RDS and DSV probabilities to be 0.02722. Thus, for 97.3% of the time, more than one transition in eight symbols will occur. It should be noted, however, that even one transition in eight is adequate for most timing recovery circuits.

A 16B9Q encoded line signal therefore contains a sufficient amount of timing information.

5.7 Bit Error Rate

A number of receiver parameters were varied in order to check the proper operation of the 16B9Q system. All tests were performed with a 16 bit input word since it was very difficult to synchronize the decoder for a pseudo random sequence, as discussed earlier. However, the results of the tests are sufficient to verify system performance. Figure 5.13 displays the BER curve obtained when the APD bias voltage was decreased. In effect, decreasing the bias voltage decreases the gain of the APD. The detector thus becomes less sensitive, since the detected photocurrent decreases, while the thermal noise voltages associated with the receiver circuit remain constant. As the APD bias voltage decreases, the signal becomes 'lost' in the thermal noise, and the BER increases. Figure 5.14 illustrates the effect of the position of the threshold level of the A/D comparators in the four level to binary decoding circuit. As long as this threshold level remains in the center of its range, the error rate is low. As it approaches the upper or lower limits of the range, however, the incoming voltage levels are more often improperly decoded since even a small amount of noise on the waveform will cause an incorrect comparison to be made. If the threshold level moves even further away from its optimum region, other voltage levels start to be improperly decoded, as one threshold level is now in the region of another. In the 16B9Q system, error free reception occurs over a central region of approximately 50 mV. As the extreme ends of this region are reached, the BER increases to 10^{-1}

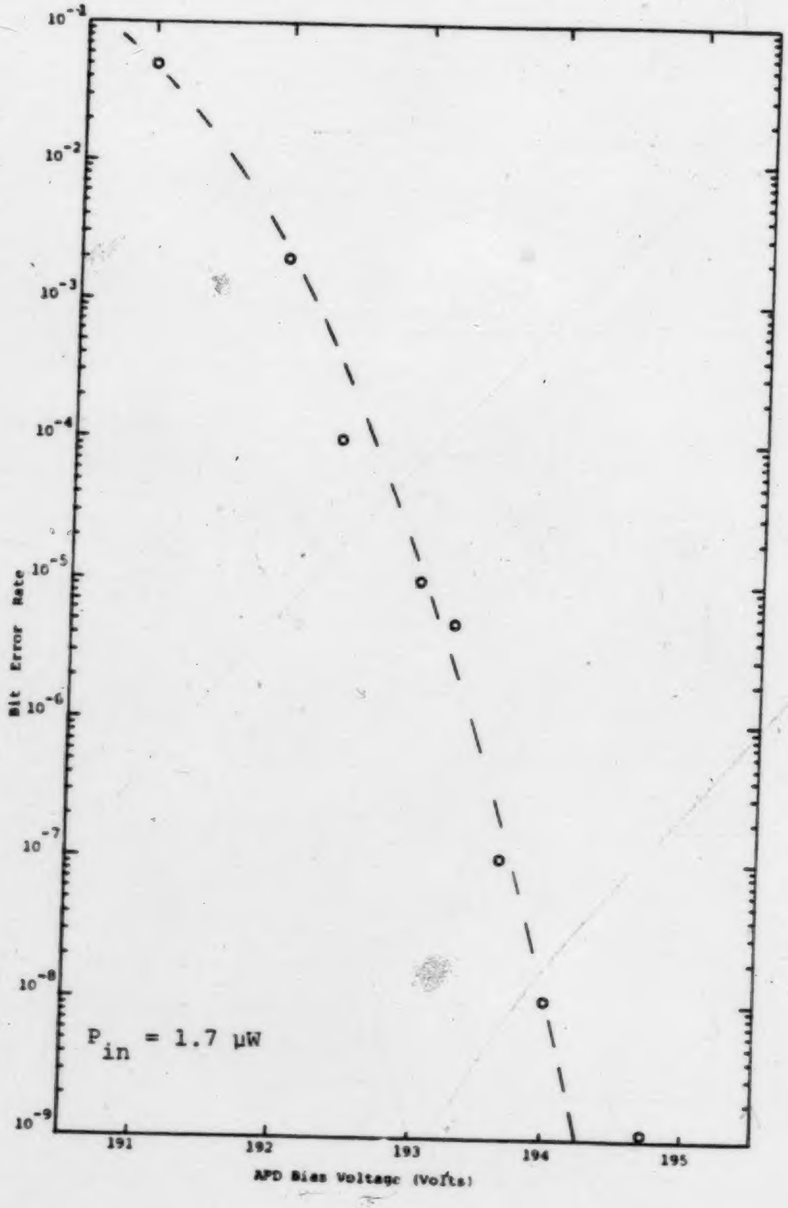


Figure 5.13. Effect of the variation of APD gain on the experimental system BER.

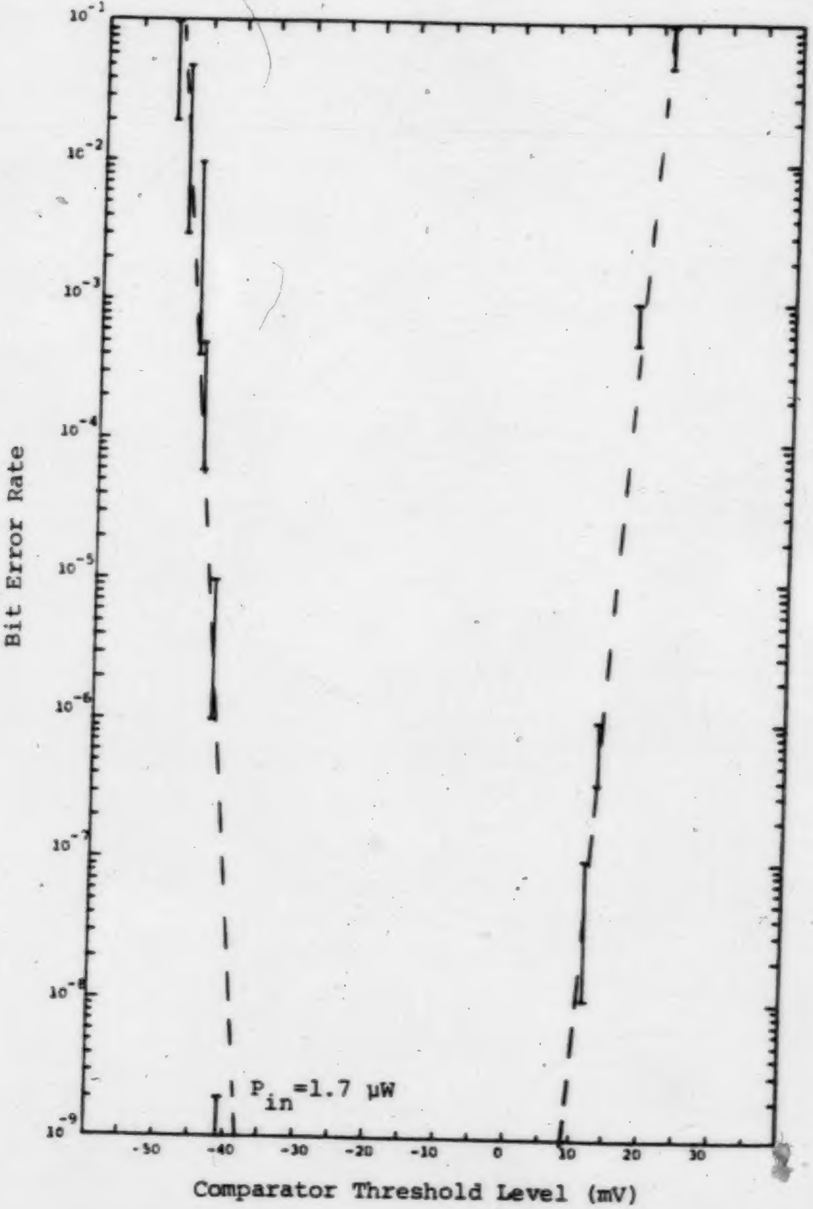


Figure 5.14. Effect of the position of comparator threshold level on the experimental system BER.

over a range of only 5 mV. The system is thus fairly sensitive to the location of its threshold levels and a good AGC amplifier is required for optimum performance. Figure 5.15 displays a typical BER curve showing the effect of variation of receiver input power. A change in input power of less than 1 dB (approximately 15%) causes the error rate to increase from 10^{-9} to 10^{-1}. It is evident that the system is quite sensitive to changes in input power. The three figures discussed above demonstrate that although some parameters of the experimental system are quite critical, it performs as expected. It is anticipated that, with a higher bandwidth preamplifier and better framing control, the system would transmit a pseudo random sequence with low bit error rates.

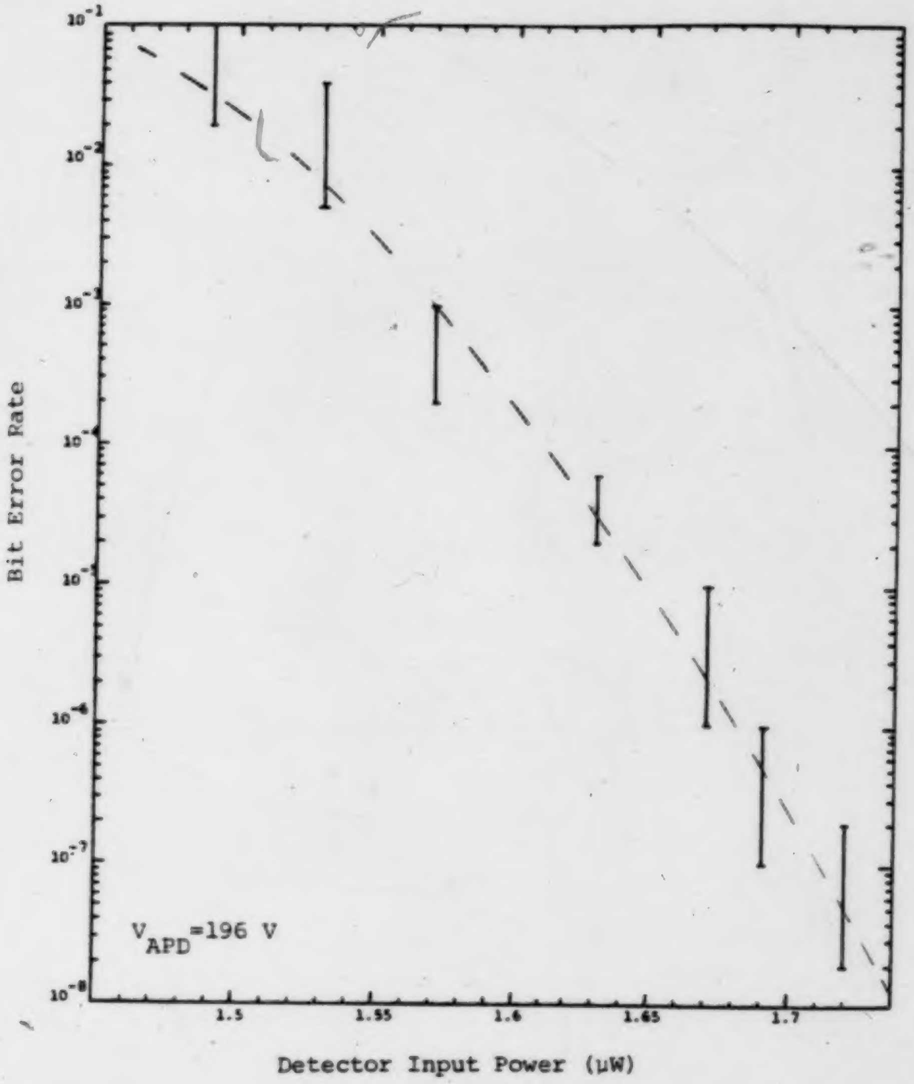


Figure 5.15. Effect of the variation of detector input power on the experimental system BER.

CHAPTER 6

SUMMARY AND CONCLUSIONS

This work has examined the applicability of the 16B9Q line code to a multimode optical fiber communication system. This code was originally developed at Bell-Northern Research (BNR) for use on short haul twisted pair copper cable systems, operating at a bit rate of 640 kb/s. The work completed in this thesis has extended the application of the code to a much higher bit rate (45 Mb/s) and to the transmission medium of optical fiber. The structures developed for the coding and decoding circuitry can be used in principle with little or no modification to implement systems operating at hundreds of megabits per second. In addition, the 16B9Q line code has been modelled as a finite state sequential machine, from which a theoretical expression for the power spectrum has been developed. The work carried out has demonstrated that this line code has many advantages and that its use results in satisfactory transmission under conditions that would cause a serious degradation in system performance to occur if a binary line code were used. It has been shown that the range of a transmission system can be extended greatly by the use of this code, that the code is bit sequence independent, and that its timing content is adequate for timing recovery.

In order for a transmitted signal to be properly received, its characteristics must be matched to those of the transmission medium; i.e., it must be able to withstand impairments caused by the channel, transmitter, and receiver with as little degradation as possible. This process of matching the signal properties to the system characteristics to facilitate optimal transmission is referred to as line coding. Several classes of line code exist: simple codes which convert one input bit into one coded symbol, multilevel codes which convert several bits into one coded symbol, and alphabetic block codes which convert a block of input bits into a block of binary or multilevel coded symbols. A new code classification, the unconventional multilevel block code, has been defined in this thesis by combining the characteristics of the multilevel and block alphabetic codes. Such a code has good spectral and timing properties, an increased information capacity, error detection capability, low error extension, bit rate and/or range extension capability, and is transparent. The encoding and decoding processes follow a set of coding rules. Therefore, the complexity of implementation does not increase greatly with increasing input block length, but remains fairly constant. The disadvantages, increased transmitter power and the necessity to consider source nonlinearity, are inherent in any multilevel code, and generally do not provide an obstacle to system implementation. The 16B9Q line code considered in this thesis is such a code.

The 16B9Q line code transforms blocks of 16 input binary digits into frames of 9 four level symbols by following rules which constrain the Running Digital Sum of the transmitted data. This is implemented by selectively inverting the multilevel words (the first half, the second half, neither or both halves of the coded multilevel block are inverted) according to the state of the polarities of the RDS and DSV. By limiting the values that the RDS can assume, and thus cause it to remain close to zero (ideally it should be zero), the DC and low frequency components of the signal are constrained. Timing information is also provided, as the maintenance of a low RDS value requires many inversions and thus many level transitions to occur.

In order to verify the various characteristics of the 16B9Q line code, an experimental system consisting of a coder, laser transmitter, multimode optical fiber, preamplifier, main amplifier and decoder was designed and constructed. The coder and decoder were designed using novel methods to perform the necessary functions of input data acceptance and storage (storage is needed in order to give the coder and decoder time to process blocks of data words while continuing to detect the incoming bit stream), calculation of the RDS and DSV, creation of the ninth symbol and its insertion into the outgoing coded data stream, and conversion of the binary to four level and four level to binary signals.

The basis of the 16B9Q line code is the selective inversion of groups of coded symbols, depending upon the value of the DSV and RDS polarities. Thus, the heart of the coder and decoder is the means by which these two sums are determined. In the original implementation of the code at BNR, this was accomplished with binary counters and pulse generating circuits. Depending on the value of the input four level symbol, either one or three pulses were generated and used to increment or decrement a binary counter. This method is satisfactory at low bit rates. However, as the bit rate increases, the limitations of the circuit components become more noticeable, until a speed is reached at which it becomes impossible to perform all the required operations in the time available. The method devised in this thesis to overcome these problems was the use of adders to calculate the DSV and RDS. Six one-bit adders were connected serially to form a six bit ripple-through adder. The DSV of the incoming word was found by serially adding the incoming bits. Results of this addition were stored in flip flops, and were then available for the RDS calculation which occurred in another six bit adder. This operation was a parallel addition, in which the entire six bit DSV was added to the previous six bit RDS. Here, the speed of circuit operation was limited only by the amount of time required to complete one six bit addition. With the components used, this time was less than 10 ns, allowing the adders to operate at frequencies exceeding 100 MHz. Thus, although the bit rate used in the experimental system was only 44.736 Mb/s, the coder and decoder could have processed a data rate of 178 MHz, and this rate could be increased simply by the use of faster components or by circuit integration.

In order to give the DSV and RDS adders sufficient time to process incoming data, a parallel storage structure was adopted in the input section of the coder. At the same time that data bits were accepted by the coder and sent to the DSV adder, they were loaded into shift registers and stored. By the time the entire 16 bit input block had been accepted, the first 8 bits were coded and ready to be sent to the transmitter. The holding time of 8 bit periods was implemented by two sets of four bit shift registers. Data was serially shifted into one shift register, which when full, performed a parallel load into the second register. An inversion signal arriving at the outputs of this second register coded the data before transmission.

A unique feature of the coder circuit implemented in this thesis was the treatment of the ninth symbol. At the start of each DSV addition, the DSV adder was initialized to a -2 at the beginning of the first word in the 16 bit input block, and to a -1 at the beginning of the second word. These initializations and subsequent inversions added the value of the ninth symbol to the RDS and DSV before that symbol had actually been created. Thus, the necessity for an extra addition to the RDS (that of the ninth symbol) was eliminated and consequently the speed of circuit operation was increased. In addition, the ninth symbol was created by the coder at a time when the entire coded word was still stored in the shift registers mentioned above. It was inserted into its proper place in the outgoing symbol stream by first moving through an additional shift register (while the first four coded symbols were transmitted) and then being added to the end of the second word as it was shifted out of the coder.

Binary to four level conversion was performed through the use of a two-to-four line decoder. The binary input digits to this device cause one of the outputs to be asserted. By differentially weighting the four outputs and summing them, a unipolar four level signal was created. At the decoder, the signal was separated into its constituent bits by first being converted to a two level signal by A/D comparators, and then being suitably manipulated so that upon presentation to the inputs of a priority encoder the resulting outputs were the original binary digits.

Upon system testing, the coder and decoder were determined to be working properly, with the system performance limited mainly by the insufficient rise time of the preamplifier. Although the preamplifier components had been modified in order to increase the bandwidth (see Appendix A) it was evident that a sufficient increase had not been obtained in practice. The spectrum of the received signal showed a large attenuation of the higher frequency components. In spite of this, the usefulness of the 16B9Q line code was verified. Successful transmission over a dispersion limited fiber showed that, for a given system bit rate, the use of this code would extend the transmission range, in accordance with theoretical predictions. Power spectra observed at the output of the coder and decoder for various pseudo random bit sequences showed that the spectrum had the same shape as the analytically derived spectrum and also as a computer generated spectrum. In addition, these spectra were identical for different input bit patterns, indicating the bit sequence independence of this line code. Finally, the bit error rate was measured in order to verify the expected system performance. Due to the difficulty of synchronizing the decoder, this was only performed for one 16 bit word, but the results of the BER tests showed that the system behaved as expected. Improvement in the preamplifier and a better framing method would allow these measurements to be performed for various pseudo random bit streams.

One of the major functions of line coding is the modification of the power spectral density of the transmitted data sequence in order to more effectively match the signal to the transmission channel characteristics. Since the coding process introduces correlations between the coded symbols which cause the statistics of the line signal to vary periodically in time (with a period equal to the length of a coded word), the coded symbol stream is a cyclostationary process. If the autocorrelation of this process is averaged over one code word period, the result is a stationary process whose Fourier transform gives the power spectrum. Although the time averaging operation causes some information to be lost, the resulting average power spectrum is nevertheless sufficient for purposes of comparing the characteristics of different line codes by comparing their spectra. Following the algorithm developed by Cariolaro and Tronca, an analytical expression for the power spectrum of the 16B9Q line code has been derived in this thesis. The 16B9Q line code was modelled as a finite state sequential machine, or Mealy machine, whose states are the values of RDS that the coder can assume, and whose next state is determined by the DSV of the incoming word and the current RDS. A transition probability matrix, $\mathbf{Q}$, which specifies the probability of going from state i to state j for i,j equal to every possible value of RDS, characterizes the coding process, and was obtained with the aid of a computer program, described in Chapter 3 and listed in Appendix C. A limiting distribution $\overset{\infty}{\bar{\mathbf{p}}}$, which specifies the absolute probability of being in a particular state and which forms the rows of the limiting matrix $\overset{\infty}{\mathbf{Q}}$ (this matrix transforms any valid initial distribution of coder states into the limiting distribution $\overset{\infty}{\bar{\mathbf{p}}}$), was found from the transition matrix. From these statistics, the power spectrum of the code was obtained. It was found that the discrete component of the 16B9Q power spectrum is zero, while the continuous component has minima at DC and at harmonics of the symbol frequency, with a fairly flat central region and small peaks in the low and high frequency regions.

A considerable potential exists for the 16B9Q line code in optical fiber communication systems. As verified experimentally, this code can be used to upgrade first generation fiber transmission systems for operation at higher bit rates. As well, it can improve the performance of other dispersion limited optical fiber systems such as long wavelength, high capacity systems(140 Mb/s or greater) which tend to be dispersion limited due to the low fiber attenuations at these wavelengths. Very high bit rate single mode systems (greater than 565 Mb/s) can also benefit from the application of the 16B9Q line code. In such systems, the high operating speed of the coding and decoding circuitry, as well as the transmitter and receiver becomes a serious limitation. The 16B9Q line code permits circuit operation at only 56% of the bit rate, thus considerably reducing design problems.

This thesis has considered the application of the 16B9Q line code to a short wavelength, multimode optical fiber system operating at a moderate bit rate of 44.736 Mb/s. The investigation was necessarily limited in scope, however, and further work remains to be done, both on the code as described in this thesis, and on new versions of the code. Possible areas of future research are outlined below:

1. Timing extraction was not considered in this thesis. It is, however, an important part of any communication system. Although theoretical considerations indicate that the 16B9Q

line code provides a more than sufficient amount of timing reference for clock extraction, this implementation could be actually carried out and the effect of jitter and noise on its performance be investigated, which could lead to better methods for clock recovery. In addition, very little work has been performed on timing extraction from a multilevel signal. Investigations in this area could be quite promising and would be widely applicable to other communication systems.

2. The possibility of a better frame finding strategy for this code could be investigated. The existing framing method relies on the knowledge of the encoder state. Although it is possible to implement state dependent decoding, the implementation of a state independent decoder is much simpler. A modification of the coding process might be needed to do this.
3. A detailed investigation of the effect of line errors on error extension and overall system performance could be carried out.
4. The 16B9Q line code was implemented in an experimental system in this thesis, without the use or consideration of any equalization. Equalization can also, however, extend the bit rate and/or range of a transmission system. The combination of line coding and equalization can therefore result in a large improvement in system performance. The potential of the 16B9Q line code together with some form of equalization could therefore be investigated.
5. This thesis has considered the application of the 16B9Q line code to dispersion limited multimode fiber systems in order to improve their bit rate and /or range. As stated earlier, however, second generation systems consisting of long wavelength sources and detectors, together with single mode fiber, are now being installed. Except for systems already in place, multimode fiber is no longer in use, and thus a bandwidth reduction code is not required since single mode fiber has a more than sufficient bandwidth for even the highest bit rates in use today. The other properties of line coding are still necessary, however, for such a system and the 16B9Q code, as well as other similar codes, contains these properties. The 16B18B line code is a modification of, and is simpler than the 16B9Q code in that the two to four level conversion is not required. The remainder of the coding procedure is identical to that of the 16B9Q code, and could be implemented using the same circuits, with the final stage (the two to four level conversion) being replaced with a multiplexer. The properties of the 16B18B line code would thus be very similar to those of

16B9Q. Any problems with source nonlinearity and the need for increased transmitter power would be alleviated, and multilevel coding and decoding would not be necessary. A variation of 16B18B is also possible. After the first word in the input block has been coded, the value of the polarity bit of the ninth symbol has already been determined. Therefore, this bit can be transmitted immediately following transmission of the first 8 bits. The magnitude bit of the ninth symbol would then be transmitted after the second 8 bits. The power spectrum of this version of 16B18B line code would be different from that of the first version, and the other code properties could also be different. It would therefore be very worthwhile to investigate these two codes and to compare their properties.

It is hoped that the work performed in this thesis will stimulate further research into the theoretical and practical aspects of line coding for optical fiber communication systems.

APPENDIX A.

DESIGN OF THE TRANSIMPEDANCE PREAMPLIFIER

A bipolar transimpedance preamplifier was used in the receiver of the experimental 16B9Q fiber optic link to perform the initial signal amplification and noise minimization following the APD detector. It consists of a cascode input stage providing good high frequency response, high current gain and low noise, followed by a common collector (CC) output stage acting as a buffer and providing impedance matching, as shown in Figure 4.9. The design of such a preamplifier was considered by El-Diwany et al. [43]; their method was applied by C. Kelly to design a 10 MHz bandwidth preamplifier [44]. The design procedure is outlined here and was used to obtain component values that increase the preamplifier bandwidth to 26 MHz.

The design procedure is concerned with the determination of the following parameters of the preamplifier (see Figure 4.9):

1. The cascode bias current I_{c2}.
2. The feedback resistor $R_f = R_4$. Since R_4 provides both ac and dc feedback, transistor Q_1 is biased by the voltage drop across R_4.
3. The common collector stage bias current I_{c3} .
4. The common collector bias resistor $R_6 = R_e$.

An ac equivalent circuit of the preamplifier is shown in Figure A.1. Replacing the transistors with their modified hybrid-pi models and performing some circuit simplification [44], the circuit of Figure A.2 is obtained. The optimum capacitance values were calculated from a computer analysis by C. Kelly [44] using the Motorola BFR90 transistor. As these transistors are used in the present preamplifier, the values are fully applicable to this situation. The cascode current of less than 1 mA is an optimum figure resulting from El-Diwany [43], and a transistor β of 50 is assumed.

Beginning the analysis at the common collector stage, the maximum output voltage of the preamplifier is first determined. For a maximum estimated optical launch power of -3 dBm and a channel attenuation of 30 dB, the maximum optical power incident on the APD is thus -33 dBm. The responsivity of the APD (RCA C30908E) is 77 A/W. A reasonable estimate for

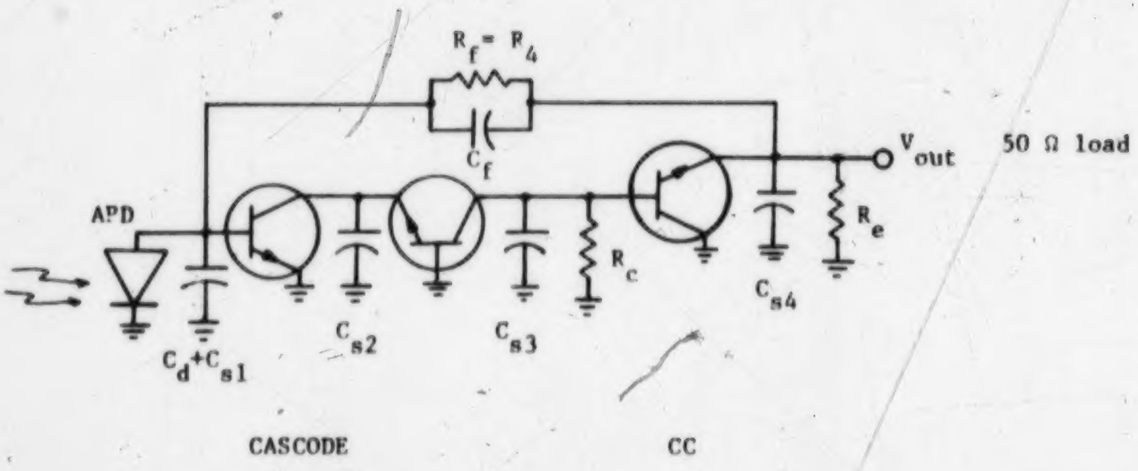
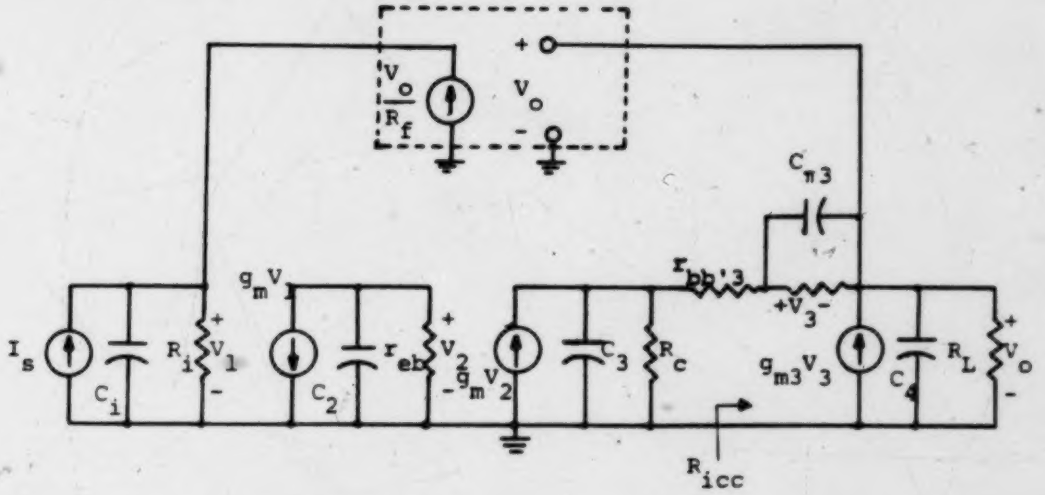


Figure A.1. AC circuit of the experimental preamplifier.



$C_i = C_d + C_\pi + C_{s1} + 2C_{\mu T} + C_f = 11.5$ pF for I_c 1 mA

$R_i = r_\pi || R_f$

$C_2 = C_{s2} + C_o + C_\pi + 9C_o = 6.5$ pF for I_c 1 mA

$r_{eb} = \beta/g_m(\beta+1) = 0.98/g_m$

$C_3 = C_{s3} + C_f + C_o = 5.5$ pF

$C_4 = C_{s4} + C_f + C_o = 11$ pF

$R_c = R_e || 50\Omega$

Figure A.2. Equivalent circuit of the preamplifier.

preamplifier transimpedance, TR, given the required signal bandwidth is 10 kΩ [43]. Therefore

$$V_{omax} = TR\cdot R_{\lambda}\cdot P_i = 1x10^4x77x10^{-6.3} = 385mV \qquad (A.1)$$

where TR is the transimpedance, R_λ is the responsivity, and P_i is the incident power. This value of V_{omax} results in a 7.7 mA peak signal current, for a 50 ohm load. A suitable bias current for the CC stage is therefore 10 mA. This value of I_{c_3} gives g_{m_3}= 385 mmhos, $r_{\pi 3}$= 130 ohms, $r_{bb'_3}$= 68 ohms, and C_{T3}= 8.5 pF.

As mentioned above, the bias voltage for Q_1 is provided by the voltage drop across R_4. If the small voltage drop across R_4 is neglected, the base-emitter voltage for Q_1 is $V_{be1} = I_{c_3}\cdot R_4$. Thus, assuming V_{be1} = 0.7 V (it could vary from 0.65 to 0.75 volts), R_4 = .7 V/10 mA = 70 ohms. An actual value of 75 ohms was used in the experimental circuit.

R_4 in parallel with 50 ohms forms the output load of the CC transistor. That is, R_L = 75||50 = 30 ohms. In order to minimize reflections at the preamplifier output, the output impedance of the CC stage should match R_L. Therefore,

$$R_{occ} = \frac{r_{\pi 3} + R_s}{\beta + 1} = 30\Omega \qquad (A.2)$$

Since the output impedance of the common base transistor, Q_2, is quite high, the source resistance, R_s, is simply R_3. Equation (A.2) then gives a value of 1400 ohms for R_3. A practical value of 1500 ohms was used in the actual implementation of the circuit.

The input impedance of the CC stage is given by

$$R_{icc} = r_{\pi 3} + R_{occ}(\beta+1) = 1.66K\Omega \qquad (A.3)$$

Thus, the resistance seen by the common base stage is

$$R_B = R_3||R_{icc} = 788\Omega \qquad (A.4)$$

The voltage gain of the CC stage can be shown to be

$$A_{vcc} = \frac{1}{1 + \frac{r_{bb'3}+z_\pi}{z_L(g_{m3}z_\pi+1)}} \qquad (A.5)$$

with $Z_\pi = r_{\pi 3} || 1/sC_3$ and $Z_L = R_L || 1/sC_4$ $(s = j\omega)$. At low frequencies,

$$A_{vcc} = \frac{1}{1 + \frac{r_{bb'3} + r_{\pi 3}}{(\beta+1)R_L}} \tag{A.6}$$

Evaluation of the frequency response of the CC stage using Equation (A.5) shows that it has a peak voltage gain of 1.05 at 800 MHz, with a 3 dB frequency of 1.39 GHz. By comparison with the desired signal bandwidth of 26 MHz, it can be assumed that this stage has a flat frequency response. This approximation allows the preamplifier circuit to be reduced to that shown in Figure A.3.

From this figure, the open loop voltage gain and transimpedance, A_0 and TR_0, of the preamplifier are found to be

$$A_o(s) = \frac{A_o}{(sT_2+1)(sT_3+1)} \tag{A.7}$$

$$TR_o(s) = \frac{A_o R_i}{(sT_1+1)(sT_2+1)(sT_3+1)} \tag{A.8}$$

where

$A_0 = 0.885 \times 0.98 \times 788 \times g_m = 683\ g_m$

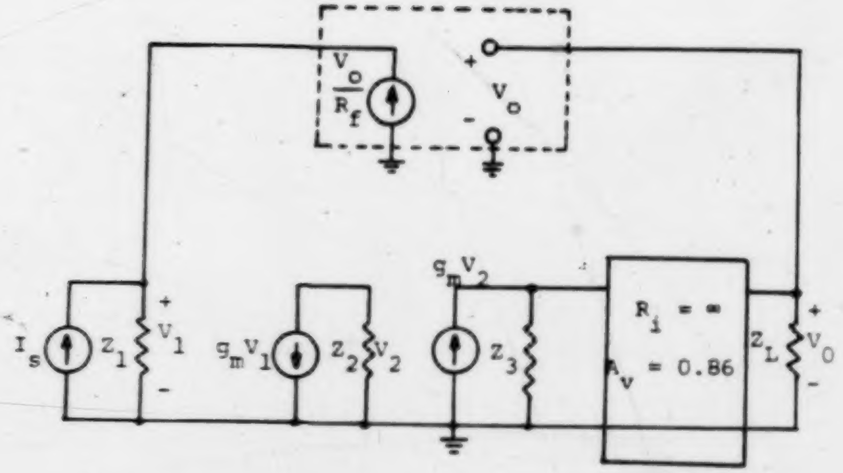
$T_1 = C_i R_i$

$T_2 = C_2 r_{eb}$

$T_3 = C_3 R_B = 5.5\ \text{pF} \times 788 = 4.3\ \text{ns}.$

If I_{c3} is less than 1 mA, then r_{eb} is less than 26 ohms and T_2 is less than .17 ns. This gives a pole at 940 MHz that can be ignored when compared to the remaining two poles, one of which occurs at 36.7 MHz (this is the one that corresponds to T_3).

The closed loop transimpedance of the preamplifier is given by



$$Z_1 = R_i || 1/sC_i = (r_\pi || R_f) || (1/s) \cdot 11.5 \text{ pF}$$

$$Z_2 = r_{eb} || 1/sC_2 = (0.98/g_m) || (1/s) \cdot 6.5 \text{ pF}$$

$$Z_3 = R_B || 1/sC_3 = 683\Omega || (1/s) \cdot 5.5 \text{ pF}$$

Figure A.3. Simplified preamplifier equivalent circuit.

$$TR_c(s) = \frac{TR_o(s)}{1 + \frac{TR_o(s)}{R_4}}$$

$$= \frac{A_oR_i}{(sT_1+1)(sT_3+1) + \frac{A_oR_i}{R_4}} \qquad (A.9)$$

which can be written in standard form as

$$TR_c(s) = \frac{TR_c}{s^2+2\xi\omega_n s+\omega_n^2} \qquad (A.10)$$

where ω_n is the natural frequency and ξ is the damping factor. Consideration of the dominant time constant T_1, and of the damping factor ξ of TR_c (s), allows an estimation of the signal bandwidth of the preamplifier to be made. The value of the damping factor is an indication of the proximity of the two dominant preamplifier poles. The larger ξ is, the further apart the poles are, and the more stable the circuit is. If the damping factor is too large, however, the circuit will have a poor rise time. To allow for a large range of stability therefore, a value of 1 is chosen for ξ (i.e., a critically damped system). For this case

$$B_s = 1.29B \quad , \quad \xi=1 \qquad (A.11)$$

where B_s is the signal bandwidth and B is the effective dominant pole. B is obtained by considering only the dominant pole of $TR_o(s)$

$$TR_{o'}(s) = \frac{A_oR_i}{sT_1+1} \qquad (A.12)$$

which yields

$$TR_c(s) = \frac{A_oR_i}{(sT_1+1) + \frac{A_oR_i}{R_4}} \qquad (A.13)$$

where B is the bandwidth of TR_c (s). This gives

$$\omega_{3dB}T = 1 + \frac{A_o R_i}{R_4}$$

$$B = \frac{1 + A_o R_i / R_4}{2\pi\ T_1} \qquad (A.14)$$

Algebraic manipulation of (A.14) yields

$$B = \frac{1}{2\pi C_i R_{eff}} \qquad (A.15)$$

where R_{eff}, the effective dominant pole resistance, is given by

$$R_{eff} = r_\pi \,||\, \frac{R_4}{1+A_o} \ . \qquad (A.16)$$

For a signal bandwidth of 26 MHz and a damping factor of 1, (A.15) gives $R_{eff} = 533$ ohms. Upon substitution of $g_m = \beta / r_\pi$ into (A.16), a relationship between R_4 and r_π is obtained:

$$533 = r_\pi \,||\, \frac{R_4}{1 + \frac{34.15K\Omega}{r_\pi}} \qquad (A.17)$$

Another relationship between these two parameters is required in order to determine their values. Such an expression can be obtained by considering the noise sources in the circuit, an extensive analysis of which has been performed by El-Diwany et al. [43]. The input referred noise for the three noise sources that can be controlled through preamplifier design (base current shot noise $S_1 = 2qI_b$, collector curren shot noise $S_2 = 2qI_c / |A_i|^2$, feedback resistor thermal noise $S_3 = 4kT/R_4$) can be shown to be [43]

$$\overline{i_n^{\ 2}} = B_n \left[\frac{4KT}{R_4} + \frac{2KT}{r_\pi} + \frac{2KTr_\pi}{\beta} \right] \left[\frac{1}{R_4^{\ 2}} + \frac{(2\pi C_i B_n)^2}{3} \right] \qquad (A.18)$$

where B_n , the noise bandwidth, can be approximated by

$$B_n = \pi/2\ B = 40.84MHz \qquad (A.19)$$

Optimum values for r_π and R_4 are then found by differentiating $\overline{i_n^2}$ with respect to r_π in Equation (A.18). This results in

$$r_{\pi OPT}^{2} = \frac{\beta + 1}{\frac{1}{R_4^2} + \frac{(2\pi C_i B_n)^2}{3}} \qquad \text{(A.20)}$$

and

$$\overline{i_{nmin}^{2}} = \left[\frac{4KT(\beta+1)B_n}{\beta} \right] \left[\frac{1}{R_4} + \frac{1}{r_\pi} \right] \qquad \text{(A.21)}$$

The second relationship between R_4 and r_π is found by substiting for C_i and B_n in (A.20)

$$r_\pi^2 = \frac{51}{\frac{1}{R_4^2} + 3.864 \times 10^{-6}} \qquad \text{(A.22)}$$

Then, assuming initial values for R_4 and r_π, and iterating with Equations (A.17) and (A.22)

$$r_{\pi opt} = 3.66 \ \text{k}\Omega$$

$$R_{4opt} = 6.9 \ \text{k}\Omega$$

are obtained. This corresponds to an optimum collector current of 0.36 mA ($I_{C_{opt}} = \beta V_T / r_\pi$) and an optimum base current of 6 μA ($I_{B_{opt}} = 2\pi C_i B_n V_T / \sqrt{3\beta}$ [43]).

Thus, for the preamplifier characteristics desired, the required resistance values are:

$R_1 = R_2 = 2700$ ohms (these simply establish the voltage at the base of Q_2)

$R_3 = 1500$ ohms (to establish collector current through Q_1 and Q_2)

$R_4 = 6900$ ohms (a potentiometer was actually used and adjusted for best results)

$R_5 = 300$ ohms (to establish the current through Q_3)

$R_6 = 75$ ohms (to bias Q_1)

APPENDIX B.

DERIVATION OF SYSTEM CONSTRAINTS

A digital optical fiber communication system, like any other data transmission system, is subject to various impairments which constrain both the information capacity and the transmission distance between repeaters. A knowledge of the behaviour of these constraints with the variation of different system parameters allows the designer to optimally implement the system. An optical fiber communication link is limited by channel dispersion, the available power at the transmitter, and the receiver sensitivity. Dispersion is determined mainly by the fiber; its effect can be modified by varying the baud rate (i.e., different coding or modulation schemes). The amount of optical power reaching the receiver is determined by the power coupled into the fiber by the source, the attenuation of the fiber, splices, and connectors, the optical power penalty due to multilevel transmission (if any), and the receiver sensitivity.

For a given system bit rate, variation of any of these parameters will change the maximum transmission distance before signal amplification and reshaping is needed. Conversely, for a given repeaterless range, a different tradeoff of these parameters will change the maximum attainable bit rate. The effect of different line codes, sources, detectors, and fibers on a system can thus be evaluated fairly easily.

The power limits are derived using

$$L = \frac{OP - M - P - R}{F + I} \tag{B.1}$$

where

L is the maximum repeater spacing

OP is the average launched optical power (3 dB less than the peak launch power of the source)

M is a loss margin, here taken to be 5 dB

P is a power penalty due to the use of any features which require more optical power, such as multilevel signalling and equalization

R is the receiver sensitivity, as given in Figure B.1 [44]

F is the fiber attenuation

I is a loss which accounts for splices, connectors, cabling, and installation. It is taken to be 0.2 dB for these calculations.

The work described in this thesis does not consider equalization at all. Therefore, the power penalty is due entirely to multilevel signalling and is given by

$$P = 10 \log_{10}(A-1) + 0.5\text{dB} \tag{B.2}$$

where A is the number of levels, and the 0.5 dB accounts for shot noise effects.

The dispersion limits are caluculated from

$$B = M \cdot f_T \tag{B 3}$$

where

M is a multiplying factor which takes into account the system baud rate ($M=1/\text{baud rate}$)

f_T is the total fiber bandwidth determined by the material and chromatic dispersion of the fiber according to $f_T = [(l/f_m)^2 + (l/f_w)^2]^{12}$, where f_m is the material dispersion, f_w is the chromatic dispersion, and l is the length of the fiber. When the total dispersion is given in ns/km (as is done by many fiber manufacturers), f_T can be calculated by $f_T = [\text{dispersion (ns/km)} \times (\text{fiber length})^7]^{-1} \times 0.441$.

Figure 1.6 was obtained using (B.1) to (B.3). The parameter varied for each curve was the line code, thus changing the optical power penalty in (B.1) and (B.2), and the multiplying factor M in (B.3). A 840 nm LED (BNR-15) was assumed as the optical source, giving a coupled power of 200 μW (-6.99 dB). The fiber was Northern Telecom Grade D with an attenuation of 3.1 dB/km and a 1.5 ns/km total dispersion. The receiver sensitivities were taken from Figure B.1, which shows the achieved receiver sensitivities for a short wavelength system using a silicon APD detector, that have been reported in the literature [44]. Table B.1 lists the attenuation limits for the five line codes displayed, while Table B.2 shows the dispersion limits.

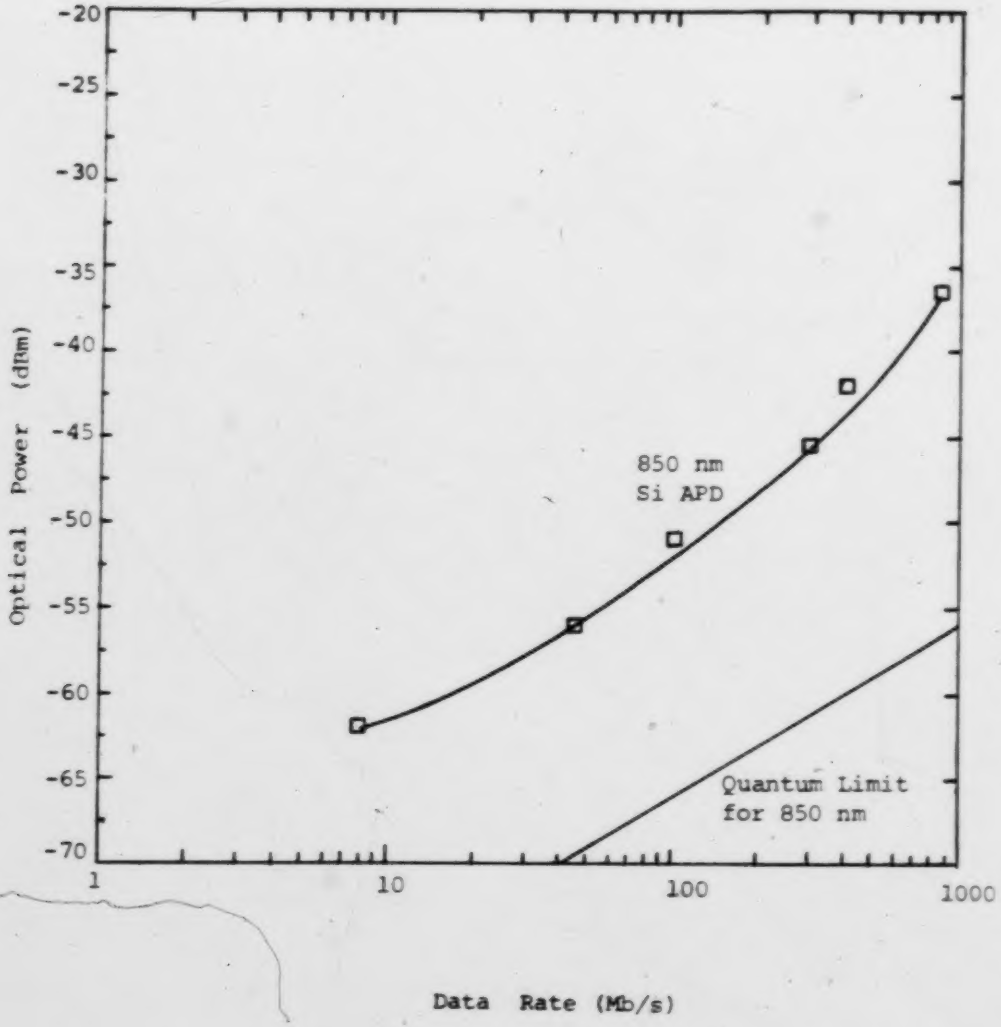


Figure B.1. Attainable receiver sensitivities for binary signalling with a 10^{-9} BER.

Table B.1. Attenuation Limits for Five Line Codes.

Baud Rate (Mb/s)	Binary (km)	2AMI (km)	4 Level PAM (km)	16B9Q (km)	8 Level PAM (km)
5	15	15	13.41	13.41	12.29
10	14.55	14.55	12.95	12.95	11.84
20	14.09	14.09	12.49	12.49	11.38
30	13.72	13.72	12.12	12.12	11.00
40	13.49	13.49	11.89	11.89	10.78
50	13.34	13.34	11.74	11.74	10.62
60	13.03	13.03	11.44	11.44	10.32
70	12.73	12.73	11.13	11.13	10.02
80	12.58	12.58	10.98	10.98	9.87
90	12.35	12.35	10.75	10.75	9.64
100	12.12	12.12	10.53	10.53	9.41
200	10.91	10.91	9.32	9.32	8.2
300	10.0	10.0	8.41	8.41	7.29
400	9.39	9.39	7.8	7.8	—
500	8.79	8.79	7.19	7.19	—

Table B.2. Dispersion Limits for Five Line Codes.

Length (km)	Binary (MHz)	2AMI (MHz)	4 Level PAM (MHz)	16B9Q (MHz)	8 Level PAM (MHz)
1	294	147	588	523.32	882
2	180.98	90.49	361.96	322.14	542.94
3	136.26	68.13	272.52	242.54	408.78
4	111.41	55.71	222.82	198.31	334.23
5	95.29	47.65	190.58	169.62	285.87
10	58.66	29.33	117.32	104.42	175.98
12.5	50.18	25.09	100.36	89.32	150.54
15	44.17	22.09	88.34	78.62	132.51
20	36.11	18.06	72.22	64.28	108.33
22.5	33.25	16.63	66.50	59.19	99.75
25	30.89	15.45	61.78	54.98	92.67
27.5	28.89	14.45	57.78	51.42	86.67
30	27.19	13.60	54.38	48.39	81.57

APPENDIX C.

COMPUTER PROGRAM LISTINGS

This appendix contains the listings of the five computer programs written to find the analytical expression for the power spectrum of the 16B9Q line code. The programs are extensively documented, and together with a knowledge of the analytical method should be easy to understand.

```
C************************************************
C
C                        PROGRAM 1
C                        ¬¬¬¬¬¬¬¬¬
C
C THIS PROGRAM CREATES THE 256 POSSIBLE 4-LEVEL 4 SYMBOL
C INPUT WORDS (0 - 255 IN BINARY), CALCULATES THEIR DSV'S,
C AND SUMS THE NUMBER OF TIMES EACH DSV OCCURS.
C THIS GIVES THE PROBABILITY DISTRIBUTION OF THE DSV VALUES
C
C*******************************************************************
      INTEGER M,BIN,T(4),SUM(13),WORD(4)
      DATA T/-1,+1,-3,+3/
C
C INITIALIZE TO 0
C
      DO 10 I=1,13
10    SUM(I)=0
C
C CREATE 4 LEVEL WORD, CALCULATE ITS DSV, AND KEEP
C A RUNNING SUM OF THE DSV'S.
C
      DO 100 BIN=1,256
      M=BIN-1
      DSV=0
      DO 50 K=1,4
      J=MOD(M,4)
      WORD(5-K)=T(J+1)
      DSV=DSV+T(J+1)
50    M=M/4
      WRITE(6,75)(WORD(L),L=1,4),DSV
100   CONTINUE
C
C WRITE OUT THE SUMS OF DSV'S
C
      DO 60 I=1,13
60    WRITE(6,85)I,SUM(I)
C
75    FORMAT(' ',4(I2),4X,I4)
85    FORMAT(' ',I2,4X,I4)
C
      STOP
      END
```

```
              C*********************************************
              C
              C                    PROGRAM 2
              C                    ¬¬¬¬¬¬¬¬¬
              C
              C THIS PROGRAM FINDS THE RDS PROBABILITIES, I.E. THE TRANSITION
              C PROBABILITY MATRIX 'Q', USING DOUBLE PRECISION VARIABLES.
              C THEN THE LIMITING TRANSITION PROBABILITY VECTOR, Q INFINITY,
              C IS DETERMINED.
              C
              C************************************************************
              C
0001                INTEGER RDSF(26),RDSI,RDS1,RDS2,DSV1,DSV2
0002                INTEGER W(13)/1,4,10,20,31,40,44,40,31,20,10,4,1/
0003                DOUBLE PRECISION Q(26,26),QI(26,26),P(26),X,WKAREA(1000)
0004                DOUBLE PRECISION D(26,26),PI(26,26)
              C
0005                DO 10 I=1,26
0006             10 RDSF(I)=0
              C
0007                DO 11 I=1,26
0008                DO 11 J=1,26
0009          11    D(I,J)=0.0D+00
              C
              C SET INITIAL VALUE OF RDS
              C
0010                DO 100 I=1,26
0011                RDSI=I-14
              C
0012                DSV1=10
              C
              C FOR EACH VALUE OF DSV1...
              C
0013                DO 200 J=1,13
              C
              C FIND THE RDS AT THE END OF THE FIRST WORD
              C
0014                IF(RDSI.LT.O.AND.DSV1.LT.O)RDS1=RDSI-DSV1
0015                IF(RDSI.GE.O.AND.DSV1.GE.O)RDS1=RDSI-DSV1
0016                IF(RDSI.LT.O.AND.DSV1.GE.O)RDS1=RDSI+DSV1
0017                IF(RDSI.GE.O.AND.DSV1.LT.O)RDS1=RDSI+DSV1
```

```
C
C THEN, FOR EVERY INITIALIZED VALUE OF DSV2 , FIND
C THE RDS AT THE END OF THE SECOND WORD.
C
      IF(RDS1.GE.0)GO TO 500
C
C RDS < 0 AND DSV > 0
C
      DO 300 K=1,11,2
      LL=(K+15)/2
      RDS2=RDS1+K
  300 RDSF(RDS2+14)=RDSF(RDS2+14)+W(J)*W(LL)
C
C RDS < 0 AND DSV < 0
C
      DO 400 K=1,13,2
      LL=(15-K)/2
      RDS2=RDS1-(-1*K)
  400 RDSF(RDS2+14)=RDSF(RDS2+14)+W(J)*W(LL)
C
      GO TO 800
C
C RDS ≥ 0 AND DSV > 0
C
  500 DO 600 K=1,11,2
      LL=(K+15)/2
      RDS2=RDS1-K
  600 RDSF(RDS2+14)=RDSF(RDS2+14)+W(J)*W(LL)
C
C RDS > 0 AND DSV < 0
C
      DO 700 K=1,13,2
      LL=(15-K)/2
      RDS2=RDS1+(-1*K)
  700 RDSF(RDS2+14)=RDSF(RDS2+14)+W(J)*W(LL)
C
C SET NEW 1ST WORD DSV
C
  800 DSV1=DSV1-2
```

```
C
  200 CONTINUE
C
C RESET RDSF ARRAY TO O FOR THE NEXT PASS
C AND STORE RDSF VALUES IN THE Q MATRIX
      DO 900 N=1,26
      Q(I,N)=RDSF(N)/0.65536D+05
  900 RDSF(N)=0
C
  100 CONTINUE
C
C ADD 1 TO EVERY MEMBER OF Q [IN EQUATION (3.33) WE ARE
C ADDING THE 'E' MATRIX TO 'Q'.  HERE 'E' IS A MATRIX
C WHOSE ENTRIES ARE ALL 1].
C
C THEN, SUBTRACT 1 FROM THE DIAGONAL ENTRIES OF 'Q+E'.
C IN EQUATION (3.33) WE ARE SUBTRACTING THE IDENTITY
C MATRIX.
C
      DO 55 J=1,26
      DO 56 K=1,26
      Q(K,J)=Q(K,J)+0.1D+01
   56 CONTINUE
   55 Q(J,J)=Q(J,J)-0.1D+01
C
C INVERT THE RESULTING MATRIX - LINV2F IS A MATRIX INVERSION
C SUBROUTINE IN THE IMSL LIBRARY ON MTS AT THE UNIVERSITY
C OF ALBERTA
C
      CALL LINV2F(Q,26,26,QI,16,WKAREA,IER)
C
C SEE IF ANY ERRORS OCCURRED DURING THE INVERSION
C
      WRITE(6,77)IER
77    FORMAT(' ','IER= ',I5)
C
      DO 45 J=1,26
      X=0.0D+00
      DO 46 K=1,26
      X=QI(K,J)+X
      P(J)=X
   46 CONTINUE
```

```
              C
              C CREATE THE 'D INFINITY' AND 'PI' MATRICES.
              C ALSO, WRITE OUT THE LIMITING TRANSITION PROBABILITY VECTOR,X.
              C
0056                WRITE(6,66)J,X
0057                D(J,J)=X
0058                PI(1,J)=X
0059             66 FORMAT('- P(',I2,' = ',D23.16)
0060             45 CONTINUE
              C
0061                DO 88 I=2,26
0062                DO 88 J=1,26
0063          88    PI(I,J)=PI(1,J)
              C
              C WRITE OUT 'D  INFINITY' AND 'PI'
              C
0064                DO 99 I=1,26
0065          99    WRITE(7,98)(D(I,J),J=1,26)
0066          98    FORMAT(' ',5(1X,D23.16))
              C
0067                DO 97 I=1,26
0068          97    WRITE(8,98)(PI(I,J),J=1,26)
              C
0069                STOP
0070                END
```

```
      C**************************************************
      C                              PROGRAM 3
      C                              ¬¬¬¬¬¬¬¬¬
      C
      C
      C THIS PROGRAM CALCULATES THE C1, C2, AND RO MATRICES FOR THE
      C 16B9Q LINE CODE. THESE MATRICES CAN THEN BE USED IN PROGRAM 5
      C TO DETERMINE THE COEFFICIENTS FOR THE ANALYTICAL POWER SPECTRUM
      C EXPRESSION.
      C
      C SINCE 65536 ITERATIONS ARE CARRIED OUT IN THIS PROGRAM, IT TAKES
      C A LONG TIME TO RUN (APPROXIMATELY 770 CPU SECONDS).
      C
      C SUBROUTINES CALLED : 1) CREATE
      C                      2) AU
      C                      3) EMULT
      C                      4) MULT
      C
      C*****************************************************************
      C
0001        INTEGER INPUT(256,4),AMATRX(26,9),EMATRX(26,26),WORD(8)
      C
0002        REAL C1(26,9),C2(26,9)
0003        REAL DTA(26,9),RO(9,9)
0004        REAL D(26,26)
      C
      C THE MAIN PROGRAM
      C
      C INITIALIZE ALL OUTPUT MATRICES TO 0
      C
0005        DO 10 I=1,26
0006        DO 10 J=1,9
0007        C1(I,J)=0.0
0008        C2(I,J)=0.0
0009  10    DTA(I,J)=0
      C
0010        DO 20 I=1,9
0011        DO 20 J=1,9
0012  20    RO(I,J)=0.0
```

```
                C
                C  READ D FROM AN INPUT FILE
                C
  0013                DO 27 I=1,26
  0014          27    READ(5,21)(D(I,J),J=1,26)
                C
  0015          21    FORMAT(' ',5(1X,E14.7))
                C
                C
                C CREATE 256 (4**4) 4 LEVEL INPUT WORDS
                C
  0016                CALL CREATE(INPUT)
                C
                C WITH THESE WORDS CREATE THE 65536 POSSIBLE INPUT WORDS.
                C CODE THEM, AND PERFORM THE NECESSARY ADDITIONS AND
                C MULTIPLICATIONS
                C
  0017                DO 300 I=1,256
                C
  0018                DO 200 L=1,256
                C
                C CREATE THE 8 SYMBOL 4-LEVEL INPUT WORD
                C
  0019                DO 100 J=1,4
  0020                WORD(J)=INPUT(I,J)
  0021          100   WORD(J+4)=INPUT(L,J)
                C
                C CREATE THE 'A' AND 'E' MATRICES FOR THE INPUT WORD
                C
  0022                CALL AU(WORD,AMATRX,EMATRX)
                C
                C INCREMENT THE 'C2' MATRIX SUM
                C
  0023                DO 110 IA=1,26
  0024                DO 110 JA=1,9
  0025          110   C2(IA,JA)=C2(IA,JA)+AMATRX(IA,JA)
                C
                C CALCULATE THE DTA (D TRANSPOSE TIMES A) MATRIX
                C FOR THIS INPUT WORD
                C
```

```
      DO 120 IA=1,26
      DO 120 JA=1,9
120   DTA(IA,JA)=D(IA,IA)*AMATRX(IA,JA)
C
C MULTIPLY 'E' BY 'DTA' AND INCREMENT THE 'C1' SUM
C
      CALL EMULT(EMATRX,DTA,C1)
C
C MULTIPLY 'A' BY 'DTA' AND INCREMENT THE 'RO' SUM
C
      CALL MULT(AMATRX,DTA,RO)
C
200   CONTINUE
300   CONTINUE
C
C SCALE THE 'C1', 'C2', AND 'RO' MATRICES BY 1/65536
C (THIS IS THE PROBABILITY OF EACH INPUT WORD - THEY
C ARE EQUIPROBABLE).
C
      DO 500 I=1,26
      DO 500 J=1,9
      C1(I,J)=C1(I,J)/65536.
500   C2(I,J)=C2(I,J)/65536.
C
      DO 600 I=1,9
      DO 600 J=1,9
600   RO(I,J)=RO(I,J)/65536.
C
C WRITE OUT THE MATRICES AND COEFFICIENTS TO A FILE
C
      WRITE(6,1001)
      DO 910 I=1,26
910   WRITE(6,1020)(C1(I,J),J=1,9)
C
      WRITE(6,1002)
      DO 920 I=1,26
920   WRITE(6,1020)(C2(I,J),J=1,9)
C
      WRITE(6,1003)
      DO 930 I=1,9
930   WRITE(6,1020)(RO(I,J),J=1,9)
```

```
         C
         C FORMAT STATEMENTS
         C
0049     1001  FORMAT('0',' **** C1 ***'//)
0050     1002  FORMAT('0',' **** C2 ****'//)
0051     1003  FORMAT('0',' **** R0 ****'//)
0052     1010  FORMAT(' ',9(1X,I2))
0053     1020  FORMAT(' ',9(1X,E12.5))
         C
0054           STOP
0055           END

         C
         C**********************************************************************
         C
0001           SUBROUTINE CREATE(INPUT)
         C
         C THIS SUBROUTINE CREATES 256 INPUT WORDS (0 - 255 IN BINARY)
         C AND PLACES THEM IN THE MATRIX 'INPUT'.
         C
0002           INTEGER M,BIN,T(4),INPUT(256,4)
0003           DATA T/-1,+1,-3,+3/
         C
         C INITIALIZE TO 0
         C
0004           DO 10 I=1,256
0005           DO 10 J=1,4
0006     10    INPUT(I,J)=0
         C
         C CREATE 4 LEVEL WORD AND STORE IT IN 'INPUT'
         C
0007           DO 100 BIN=1,256
0008           M=BIN-1
0009           DO 50 K=1,4
0010           J=MOD(M,4)
0011           INPUT(BIN,5-K)=T(J+1)
0012     50    M=M/4
0013     100   CONTINUE
         C
0014           RETURN
0015           END
```

```
C
C******************************************************************
C
      SUBROUTINE AU(WORD,AMATRX,EMATRX)
C
C FOR AN INPUT WORD, THIS SUBROUTINE CREATES A MATRIX OF THE
C OUTPUT WORDS IN EACH CODER STATE (THE 'A' MATRIX), AS WELL
C AS THE 'E' MATRIX FOR THAT WORD.
C
      INTEGER WORD(8),AMATRX(26,9),EMATRX(26,26)
      INTEGER RDS1,RDS2,DSV1,DSV2,STATE
C
C INITIALIZE TO 0
C
      DO 10 I=1,26
      DO 10 J=1,9
10    AMATRX(I,J)=0
C
      DO 15 I=1,26
      DO 15 J=1,26
15    EMATRX(I,J)=0
C
C INITIALIZE THE FIRST WORD AND SECOND WORD DSV'S
C
      DSV1=-2
      DSV2=-1
C
C CALCULATE THE FIRST AND SECOND WORD DSV'S
C NOTE: THESE ARE INITIALIZED DSV'S
C
      DO 20 J=1,4
      DSV1=DSV1+WORD(J)
20    DSV2=DSV2+WORD(J+4)
C
C FOR EACH POSSIBLE RDS VALUE (I.E., FOR EACH POSSIBLE
C STATE OF THE CODER),
C DETERMINE WHETHER THE FIRST WORD, SECOND WORD, NEITHER,
C OR BOTH NEED TO BE INVERTED BEFORE TRANSMISSION, BY
C USING THE 16B9Q CODING RULES.
C
C THAT IS, GIVEN AN INPUT WORD, FIND THE OUTPUT WORD
C FOR EVERY CODER STATE. THIS IS THE 'A' MATRIX.
C
```

```
204.000     0015            DO 100 I=1,26
205.000                C
206.000                C SET THE RDS VALUE (CODER STATE)
207.000                C
208.000     0016            IRDS=I-14
209.000                C
210.000                C FIND THE RDS AT THE END OF THE FIRST WORD
211.000                C
212.000     0017            IF(IRDS.LT.0.AND.DSV1.LT.0)RDS1=IRDS-DSV1
213.000     0018            IF(IRDS.GE.0.AND.DSV1.GE.0)RDS1=IRDS-DSV1
214.000     0019            IF(IRDS.LT.0.AND.DSV1.GE.0)RDS1=IRDS+DSV1
215.000     0020            IF(IRDS.GE.0.AND.DSV1.LT.0)RDS1=IRDS+DSV1
216.000                C
217.000                C DETERMINE WHETHER AN INVERSION IS NEEDED
218.000                C
219.000     0021            IF(IRDS.LT.0.AND.DSV1.LT.0)INV1=-1
220.000     0022            IF(IRDS.GE.0.AND.DSV1.GE.0)INV1=-1
221.000     0023            IF(IRDS.LT.0.AND.DSV1.GE.0)INV1=1
222.000     0024            IF(IRDS.GE.0.AND.DSV1.LT.0)INV1=1
223.000                C
224.000                C FIND THE RDS AT THE END OF THE SECOND WORD
225.000                C
226.000     0025            IF(RDS1.LT.0.AND.DSV2.LT.0)RDS2=RDS1-DSV2
227.000     0026            IF(RDS1.GE.0.AND.DSV2.GE.0)RDS2=RDS1-DSV2
228.000     0027            IF(RDS1.LT.0.AND.DSV2.GE.0)RDS2=RDS1+DSV2
229.000     0028            IF(RDS1.GE.0.AND.DSV2.LT.0)RDS2=RDS1+DSV2
230.000                C
231.000                C DETERMINE WHETHER AN INVERSION IS NEEDED
232.000                C
233.000     0029            IF(RDS1.LT.0.AND.DSV2.LT.0)INV2=-2
234.000     0030            IF(RDS1.GE.0.AND.DSV2.GE.0)INV2=-2
235.000     0031            IF(RDS1.LT.0.AND.DSV2.GE.0)INV2=2
236.000     0032            IF(RDS1.GE.0.AND.DSV2.LT.0)INV2=2
237.000                C
238.000                C FROM THE INFORMATION THAT WAS FOUND ABOVE,
239.000                C CREATE THE CODED WORD AND FIND THE VALUE OF THE NINTH SYMBOL
240.000                C
241.000     0033            STATE=INV1+INV2
242.000                C
243.000     0034            IF(STATE.EQ.-3)GO TO 30
```

```
      IF(STATE.EQ.1)GO TO 40
      IF(STATE.EQ.-1)GO TO 50
      IF(STATE.EQ.3)GO TO 60
C
C INVERT THE WHOLE BLOCK - NINTH SYMBOL=3
C
30    DO 35 K=1,8
35    AMATRX(I,K)=-1*WORD(K)
      AMATRX(I,9)=3
      GO TO 70
C
C INVERT THE FIRST WORD - NINTH SYMBOL =1
C
40    DO 45 K=1,4
      AMATRX(I,K)=-1*WORD(K)
45    AMATRX(I,K+4)=WORD(K+4)
      AMATRX(I,9)=1
      GO TO 70
C
C INVERT THE SECOND WORD - NINTH SYMBOL=-1
C
50    DO 55 K=1,4
      AMATRX(I,K)=WORD(K)
55    AMATRX(I,K+4)=-1*WORD(K+4)
      AMATRX(I,9)=-1
      GO TO 70
C
C LEAVE THE WHOLE BLOCK UNCHANGED - NINTH SYMBOL=
C
60    DO 65 K=1,8
65    AMATRX(I,K)=WORD(K)
      AMATRX(I,9)=-3
C
C PUT A '1' IN THE 'E' MATRIX IN THE COLUMN CORRESPONDING
C TO THE RDS AT THE END OF THE CODED WORD (THE NEXT STATE)
C
70    EMATRX(I,RDS2+14)=1
C
100   CONTINUE
C
      RETURN
      END
```

```
C
C***********************************************************************
C
      SUBROUTINE EMULT(EMATRX,DTA,C1)
C
C THIS SUBROUTINE MULTIPLIES THE DTA MATRIX BY THE TRANSPOSE OF
C EMATRX AND THEN ADDS THIS TO THE RUNNING SUM IN THE C1
C MATRIX.
C
      REAL DTA(26,9),C1(26,9),CCTEMP(26,9)
      INTEGER EMATRX(26,26)
C
C INITIALIZE TO O
C
      DO 10 I=1,26
      DO 10 J=1,9
10    CCTEMP(I,J)=0.0
C
C MULTIPLY EMATRX TRANSPOSE BY Q AND STORE IN CCTEMP
C
      DO 100 I=1,26
      DO 100 J=1,9
      DO 100 K=1,26
100   CCTEMP(I,J)=CCTEMP(I,J)+EMATRX(K,I)*DTA(K,J)
C
C ADD CCTEMP TO CC
C
      DO 200 I=1,26
      DO 200 J=1,9
200   C1(I,J)=C1(I,J)+CCTEMP(I,J)
C
      RETURN
      END
```

```
C
C***********************************************************************
C
      SUBROUTINE MULT(AA,BB,CC)
C
C THIS SUBROUTINE MULTIPLIES THE BB MATRIX BY THE TRANSPOSE OF
C THE AA MATRIX AND THEN ADDS THIS TO THE RUNNING SUM IN THE CC
C MATRIX.
C
      REAL BB(26,9),CC(9,9),CCTEMP(9,9)
      INTEGER AA(26,9)
C
C INITIALIZE TO 0
C
      DO 10 I=1,9
      DO 10 J=1,9
10    CCTEMP(I,J)=0.0
C
C MULTIPLY AA TRANSPOSE BY BB AND STORE IN CCTEMP
C
      DO 100 I=1,9
      DO 100 J=1,9
      DO 100 K=1,26
100   CCTEMP(I,J)=CCTEMP(I,J)+AA(K,I)*BB(K,J)
C
C ADD CCTEMP TO CC
C
      DO 200 I=1,9
      DO 200 J=1,9
200   CC(I,J)=CC(I,J)+CCTEMP(I,J)
C
      RETURN
      END
```

```
C*************************************************
C
C                     PROGRAM 4
C                     ¬¬¬¬¬¬¬¬¬
C
C USING THE C1, C2, AND RO MATRICES CALCULATED IN PROGRAM 3,
C THIS PROGRAM CALCULATES THE COEFFICIENTS FOR THE ANALYTICAL
C POWER SPECTRUM EXPRESSION FOR THE 16B9Q LINE CODE.
C
C
C SUBROUTINES CALLED:
C                      1) RINF
C                      2) DANDG
C                      3) HKAY
C                      4) NS
C                      5) NUM
C                      6) DQ
C                      7) MU
C                      8) NU
C
C**********************************************************************
C
      REAL C1(26,9),C2(26,9)
      REAL RINFIN(9,9),RO(9,9),G(702,26),H(234,9)
      REAL PI(26,26),UPI(26,26),F(26,26)
      REAL DD(27),MUK(9),NUK(9),DEQ(27),NPKN(243),NUMER(243)
C
C THE MAIN PROGRAM
C
C  READ THE PI, UPI, AND F MATRICES FROM A FILE
C
C PI  = THE Q INFINITY MATRIX
C UPI = THE IDENTITY MATRIX - THE Q INFINITY MATRIX
C F   = THE Q MATRIX - THE Q INFINITY MATRIX
C
      DO 25 I=1,26
 25   READ(5,21)(PI(I,J),J=1,26)
C
      DO 26 I=1,26
 26   READ(5,21)(UPI(I,J),J=1,26)
```

```
C
      DO 28 I=1,26
28    READ(5,21)(F(I,J),J=1,26)
C
21    FORMAT(' ',5(1X,E14.7))
C
C READ C1,C2 AND RO FROM FILES
C
      DO 32 I=1,26
32    READ(7,1020)(C1(I,J),J=1,9)
C
      DO 33 I=1,26
33    READ(7,1020)(C2(I,J),J=1,9)
C
      DO 34 I=1,9
34    READ(7,1020)(RO(I,J),J=1,9)
C
C CALCULATE THE 'R INFINTITY' MATRIX
C
      CALL RINF(C1,PI,C2,RINFIN)
C
C CALCULATE THE 'D' VECTOR AND THE 'G' MATRICES
C
      CALL DANDG(F,DD,G)
C
C CALCULATE THE 'H' MATRICES
C
      CALL HKAY(C1,UPI,G,C2,H)
C
C CALCULATE THE 'NS' NUMERATOR COEFFICIENTS
C
      CALL NS(H,NPKN)
C
C CALCULATE THE FINAL 'NK' NUMERATORE COEFFICIENTS
C (NK = NPKN HERE)
C
      CALL NUM(DD,NPKN,NUMER)
C
C CALCULATE THE 'DK' DENOMINATOR COEFFICIENTS
C (DK = DEQ HERE)
C
```

```
0023              CALL DQ(DD,DEQ)
            C
            C CALCULATE THE MU COEFFICIENTS
            C
0024              CALL MU(RINFIN,MUK)
            C
            C CALCULATE THE NU COEFFICIENTS
            C
0025              CALL NU(RO,RINFIN,NUK)
            C
            C WRITE OUT THE MATRICES AND COEFFICIENTS
            C
0026              WRITE(6,1001)
0027              DO 910 I=1,26
0028        910   WRITE(6,1020)(C1(I,J),J=1,9)
            C
0029              WRITE(6,1002)
0030              DO 920 I=1,26
0031        920   WRITE(6,1020)(C2(I,J),J=1,9)
            C
0032              WRITE(6,1003)
0033              DO 930 I=1,9
0034        930   WRITE(6,1020)(RO(I,J),J=1,9)
            C
0035              WRITE(6,1004)
0036              DO 940 I=1,9
0037        940   WRITE(6,1020)(RINFIN(I,J),J=1,9)
            C
0038              WRITE(6,1005)
0039              DO 950 I=1,702
0040        950   WRITE(6,1030)(G(I,J),J=1,26)
            C
0041              WRITE(6,1006)
0042              WRITE(6,1040)(DD(I),I=1,27)
            C
0043              WRITE(6,1007)
0044              DO 960 I=1,234
0045        960   WRITE(6,1020)(H(I,J),J=1,9)
            C
```

```
      WRITE(6,1008)
      WRITE(6,1040)(NPKN(I),I=1,243)
C
      WRITE(6,1009)
      WRITE(6,1040)(NUMER(I),I=1,243)
C
      WRITE(6,1011)
      WRITE(6,1040)(DEQ(I),I=1,27)
C
C FORMAT STATEMENTS
C
1001  FORMAT('O',' **** C1 ***'//)
1002  FORMAT('O',' **** C2 ****'//)
1003  FORMAT('O',' **** RO ****'//)
1004  FORMAT('O',' **** RINFINITY ****'//)
1005  FORMAT('O',' **** G ****'//)
1006  FORMAT('O',' **** D ****'//)
1007  FORMAT('O',' **** H ****'//)
1008  FORMAT('O',' **** NPKN ****'//)
1009  FORMAT('O',' **** NUM ****'//)
1011  FORMAT('O',' **** DEN ****'//)
C
1000  FORMAT('O',' WORD ',I5,1X ,': ',8(1X,I2))
1010  FORMAT(' ',9(1X,I2))
1020  FORMAT(' ',9(1X,E12.5))
1030  FORMAT(' ',10(2X,E12.5))
1040  FORMAT(' ',10(2X,E12.5))
C
      STOP
      END
```

```
                C
                C****************************************************************
                C
0001                  SUBROUTINE RINF(C1,PI,C2,RINFIN)
                C
                C THE RINFINITY MATRIX IS CALCULATED BY MULTIPLYING THE TRANSPOSE
                C OF 'C1' BY 'PI' BY 'C2'.
                C
0002                  REAL C1(26,9),C2(26,9),PI(26,26),RINFIN(9,9),RTEMP(9,26)
                C
                C INITIALIZE TO 0
                C
0003                  DO 10 I=1,9
0004                  DO 10 J=1,9
0005            10    RINFIN(I,J)=0.0
                C
0006                  DO 20 I=1,9
0007                  DO 20 J=1,26
0008            20    RTEMP(I,J)=0.0
                C
                C MULTIPLY THE TRANSPOSE OF 'C1' BY 'PI'
                C
0009                  DO 100 I=1,9
0010                  DO 50 J=1,26
0011                  DO 50 K=1,26
0012            50    RTEMP(I,J)=RTEMP(I,J)+C1(K,I)*PI(K,J)
0013            100   CONTINUE
                C
                C MULTIPLY THE RESULT OF THE ABOVE BY 'C2'
                C
0014                  DO 200 I=1,9
0015                  DO 150 J=1,9
0016                  DO 150 K=1,26
0017            150   RINFIN(I,J)=RINFIN(I,J)+RTEMP(I,K)*C2(K,J)
0018            200   CONTINUE
                C
0019                  RETURN
0020                  END
```

```
C
C******************************************************************
C
      SUBROUTINE DANDG(F,DD,G)
C
C  THE 'D' VECTOR AND THE 'G' MATRIX ARE DETERMINED RECURSIVELY,
C  USING THE 'F' MATRIX AS INPUT (F=PI-PIINFINITY).
C
      REAL DD(27),G(702,26),F(26,26),FG(26,26),DU(26,26)
C
C SET THE FIRST 'G' MATRIX TO BE THE IDENTITY MATRIX
C & INITIALIZE THE OTHERS TO O
C
      DO 10 I=1,26
      DO 10 J=1,26
      DU(I,J)=0.0
      G(I,J)=0.0
10    IF(I.EQ.J)G(I,J)=1.0
C
C SET THE FIRST VALUE OF 'DD' TO 1
C
      DD(1)=1.
C
      DO 100 I=2,27
      J=26*I-26
C
      DO 20 I1=1,26
      DO 20 J1=1,26
20    FG(I1,J1)=0.0
C
C F TIMES G(K-1)
C
      DO 40 K=1,26
      DO 30 L=1,26
      J1=J-26
      DO 30 LL=1,26
      J1=J1+1
30    FG(K,L)=FG(K,L)+F(K,LL)*G(J1,L)
40    CONTINUE
C
      TRACE=0.0
```

```
C
      DO 50 K=1,26
50    TRACE=TRACE+FG(K,K)
C
      DD(I)=-1*TRACE/(I-1)
C
C MULTIPLY THE KTH ELEMENT OF  'DD' BY THE IDENTITY MATRIX
C
      DO 70 K=1,26
70    DU(K,K)=DD(I)
C
C FIND G(K)
C
      DO 80 K=1,26
      J=J+1
      DO 80 L=1,26
80    G(J,L)=FG(K,L)+DU(K,L)
C
100   CONTINUE
C
      RETURN
      END
```

```
C
C**********************************************************************
C
      SUBROUTINE HKAY(C1,UPI,G,C2,H)
C
C THIS SUBROUTINE CALCULATES THE 'H' MATRICES
C
      REAL C1(26,9),C2(26,9),UPI(26,26),G(702,26),H(234,9)
      REAL HTEMP(9,26),C1UPI(9,26)
C
C INITIALIZE TO 0
C
      DO 10 I=1,234
      DO 10 J=1,9
10    H(I,J)=0.0
C
      DO 20 I=1,9
      DO 20 J=1,26
20    C1UPI(I,J)=0.0
C
C MULTIPLY THE TRANSPOSE OF 'C1' BY UPI (IDENTITY MATRIX - PI)
C
      DO 100 I=1,9
      DO 50 J=1,26
      DO 50 K=1,26
50    C1UPI(I,J)=C1UPI(I,J)+C1(K,I)*UPI(K,J)
100   CONTINUE
C
C MULTIPLY THE ABOVE BY 'G(25-K)' & 'C2'
C
      DO 500 I=1,26
      J=702-26*I
      JJ=9*I-9
C
      DO 125 II=1,9
      DO 125 III=1,26
125   HTEMP(II,III)=0.0
C
C MULTIPLY BY 'G(25-K)'
C
```

```
292.000    0021              DO 200 K=1,9
293.000    0022              DO 150 L=1,26
294.000    0023              J1=J
295.000    0024              DO 150 LL=1,26
296.000    0025              J1=J1+1
297.000    0026        150   HTEMP(K,L)=HTEMP(K,L)+C1UPI(K,LL)*G(J1,L)
298.000    0027        200   CONTINUE
299.000                C
300.000                C MULTIPLY THE ABOVE BY 'C2'
301.000                C
302.000    0028              DO 300 K=1,9
303.000    0029              JJ=JJ+1
304.000    0030              DO 250 L=1,9
305.000    0031              DO 250 LL=1,26
306.000    0032        250   H(JJ,L)=H(JJ,L)+HTEMP(K,LL)*C1(LL,L)
307.000    0033        300   CONTINUE
308.000                C
309.000    0034        500   CONTINUE
310.000                C
311.000    0035              RETURN
312.000    0036              END
```

```
C
C*****************************************************************
C
      SUBROUTINE NS(H,NPKN)
C
C  THIS SUBROUTINE CALCULATES THE NUMERATOR COEFFICIENTS NPKN.
C
      REAL H(234,9),NPKN(243),NTEMP,NTEMP2
C
C INITIALIZE TO 0
C
      DO 10 I=1,243
10    NPKN(I)=0.0
C
      DO 500 K=1,27
      IJ=9*K-18
C
      NTEMP=0.0
      NTEMP2=0.0
C
      DO 100 IP=1,9
      J=9*K+IP-9
      ISUM=10-IP
      IJ1=IJ
C
      DO  50 I=1,ISUM
      IF(K.EQ.1)GO TO 55
      IJ1=IJ1+1
50    NTEMP=NTEMP+H(IJ1+IP-1,I)
C
      GO TO 60
55    NTEMP=0.0
C
```

```
346.000     0019        60    IJ1=IJ+9
347.000     0020              IP2=IP-1
348.000     0021              DO 75 I=1,IP2
349.000     0022              IF(K.EQ.27)GO TO 80
350.000     0023              IJ1=IJ1+1
351.000     0024              IF(I.EQ.1)NTEMP2=NTEMP2
352.000     0025        75    IF(I.GT.1)NTEMP2=NTEMP2+H(IJ1,I+10-IP)
353.000                 C
354.000     0026              GO TO 90
355.000     0027        80    NTEMP2=0.0
356.000                 C
357.000     0028        90    CONTINUE
358.000                 C
359.000     0029        100   NPKN(J)=NTEMP+NTEMP2
360.000                 C
361.000     0030        500   CONTINUE
362.000                 C
363.000     0031              RETURN
364.000     0032              END
```

```
                  C
                  C***********************************************************************
                  C
0001                    SUBROUTINE NUM(DD,NPKN,NUMER)
                  C
                  C THIS SUBROUTINE CALCULATES THE NUMERATOR COEFFICIENTS FOR X2(F)
                  C
0002                    REAL DD(27),NPKN(243),NUMER(243)
                  C
                  C INITIALIZE TO O
                  C
0003                    DO 10 I=1,243
0004              10    NUMER(I)=0.0
                  C
0005                    DO 200 IP=1,9
0006                    DO 100 IQ=1,27
0007                    ISUM=28-IQ
0008                    DO 50 IH=1,ISUM
0009              50    NUMER(IP+9*IQ-9)=NUMER(IP+9*IQ-9)+DD(28-IH)*NPKN((IH+IQ-2)*9+IP)+
                       *DD(29-IH-IQ)*NPKN(9*IH-IP-8)
                  C
0010              100   CONTINUE
0011              200   CONTINUE
                  C
0012                    RETURN
0013                    END
```

```
C
C******************************************************************
C
      SUBROUTINE DQ(DD,DEQ)
C
C THIS SUBROUTINE CALCULATES THE DENOMINATOR COEFFICIENTS OF X2(F)
C
      REAL DD(27),DEQ(27)
C
C INITIALIZE TO O
C
      DO 10 I=1,27
10    DEQ(I)=0.0
C
      DO 100 IQ=1,27
      ISUM=28-Q
      DO 50 IH=1,ISUM
50    DEQ(IQ)=DEQ(IQ)+DD(28-IH)*DD(29-IH-IQ)
100   CONTINUE
C
      RETURN
      END
```

```
             C
             C*******************************************************************
             C
      0001         SUBROUTINE MU(RINFIN,MUK)
             C
             C THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR THE DISCRETE SPECTRUM
             C
      0002         REAL RINFIN(9,9),MUK(9)
             C
             C INITIALIZE TO O
             C
      0003         DO 10 I=1,9
      0004   10    MUK(I)=0.0
             C
      0005         DO 100 K=1,9
      0006         ISUM=10-K
      0007         DO 50 I=1,ISUM
      0008   50    MUK(K)=MUK(K)+RINFIN(I,I+K-1)
      0009   100   CONTINUE
             C
      0010         RETURN
      0011         END
```

```
C
C*******************************************************************
C
      SUBROUTINE NU(RO,RINFIN,NUK)
C
C THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR X1(F)
C
      REAL RO(9,9),RINFIN(9,9),NUK(9)
C
C INITIALIZE TO O
C
      DO 10 I=1,9
10    NUK(I)=0.0
C
      DO 100 K=1,9
      ISUM=10-K
      DO 50 I=1,ISUM
50    NUK(K)=NUK(K)+RO(I,I+K-1)-RINFIN(I,I+K-1)
100   CONTINUE
C
      RETURN
      END
```

```
C**********************************************
C
C                 PROGRAM 5
C                 ¬¬¬¬¬¬¬¬¬
C
C THIS PROGRAM EVALUATES THE EXPRESSION FOR POWER SPECTRUM
C
C*********************************************************
C
      DOUBLE PRECISION NUM(243),DEN(27),NU(9),NUMER,DENOM
      DOUBLE PRECISION X1,XC,ARG
      REAL FREQ
C
      PI=3.141592654
      CON=2*PI*39.73931E-09
C
C READ THE COEFFICIENTS FROM FILES
C
      READ(5,1000)(NUM(I),I=1,243)
      READ(5,1000)(DEN(I),I=1,27)
      READ(5,1000)(NU(I),I=1,9)
C
      DO 500 L=1,86
C
C READ THE FREQUENCY FROM A FILE
C
      READ(8,2000)FREQ
C
C EVALUATE THE EXPRESSION
C
      X1=0.OD+00
      NUMER=0.OD+00
      DENOM=0.OD+00
C
      DO 100 I=1,8
      ARG=CON*FREQ*I
100   X1=X1+NU(I+1)*DCOS(ARG)
      X1=2*X1+NU(1)
C
      DO 200 I=1,242
      ARG=CON*FREQ*I
200   NUMER=NUMER+NUM(I+1)*DCOS(ARG)
      NUMER=2*NUMER+NUM(1)
```

```
C
      DO 300 I=1,26
      ARG=CON*9*FREQ*I
300   DENOM=DENOM+DEN(I+1)*DCOS(ARG)
      DENOM=2*DENOM+DEN(1)
C
      XC=X1+NUMER/DENOM
C
      WRITE(6,1010)FREQ,XC
      WRITE(7,1020)X1,NUMER,DENOM,XC
C
500   CONTINUE
C
600   CONTINUE
C
1000  FORMAT(' ',5(1X,D23.16))
1010  FORMAT(' ',2(2X,D23.16))
1020  FORMAT(' ',4(1X,D23.16))
2000  FORMAT(' ',1X,E14.7)
C
      STOP
      END
```

BIBLIOGRAPHY

1. J. Gowar, **Optical Communication Systems**, Prentice-Hall International Inc., London, 1984.
2. E. L. Safford Jr., **The Fiber Optics and Laser Handbook**, TAB BOOKS Inc., Blue Ridge Summit PA., 1984.
3. J. Hecht, "Victorian Experiments and Optical Communications", **IEEE Spectrum**, Vol. 22, No. 2, February 1985, pp 69-73.
4. T. Li., "Advances in Optical Fiber Communications: An Historical Perspective", **IEEE Journal on Selected Areas in Communications** , Vol. SAC-1, No. 3, April 1983, pp 356-372.
5. Staff of Bell-Northern Research and Northern Telecom Canada, **Course Notes on Optical Fiber Communications and Technology** , 1984.
6. D. J. Morris, **Pulse Code Formats For Fiber Optical Data Communication**, Marcel Dekker Inc., New York, 1983.
7. G. Keiser, **Optical Fiber Communications**, McGraw-Hill Inc., New York, 1983.
8. R. M. Brooks and A. Jessop, "Line Coding for Optical Fiber Systems", **International Journal of Electronics**, Vol. 55, No. 1, 1983, pp 82-120.
9. P. Bylanski and D. G. W. Ingram, **Digital Transmission Systems**, Peter Peregrinus Ltd., England, 1976.
10. K. W. Cattermole and J. J. O'Reilly, **Problems of Randomness in Communication Engineering**, Pentech Press Ltd., 1984.
11. T. V. Muoi and J. L Hullet,"Receiver Design for Multilevel Digital Optical Fiber Systems", **IEEE Transactions on Communications**, September 1975, pp 987-994.
12. R. Petrovic, "Multilevel Signals in Digital Optical Fiber Communications", **Electronics Letters**, Vol. 14, No. 25, December 1978, pp 806-808.
13. H. Kressel : Editor, **Semiconductor Devices for Optical Communications**, Vol. 39, Springer-Verlag, New York, 1980.
14. L. Bosotti and G. Tamburelli, "Performance Evaluation in Multilevel Digital Transmission on Optical Fibers", **Proceedings of the 4th European Conference on Optical Communications (ECOC)**, Genoa, Italy, September 1978, pp 539-545.
15. D. B. Waters, "Codes for Digital Line Transmission", **Communications International**, Vol.

5, No. 3, March 1978, pp 19-27.

16. J. M. Sipress, "A New Class of Selected Ternary Pulse Transmission Plans for Digital Transmission Lines", **IEEE Transactions on Communication Technology**, Vol. 13, No. 3, September 1965, pp 366-372.
17. A. Jessop, D. B. Waters, "4B-3T, An Efficient Code for PCM Coaxial Line Systems", **Proceedings 7th International Scientific Congress on Electronics**, March 1970, Rome, pp 277-283.
18. J. B. Buchner, "Ternary Line Signal Codes", **Philips Telecommunications Review**, Vol. 34, No. 2, June 1976, pp 72-86.
19. A. Croisier, "Introduction to Pseudoternary Transmission Codes", **IBM Journal of Research and Development**, July 1970, pp 354-376.
20. O. Brugia, M. Decina, and U. de Julio, "Character Alignment for High-Capacity PCM Systems Using MS43 Line Code", **IEEE Transactions on Communications**, Vol. COM-23, No. 8, August 1975, pp 803-812.
21. R. C. Houtts and T. A. Green, "Comparing Bandwidth Requirements for Binary Baseband Signals", **IEEE Transactions on Communications**, June 1973, pp 776-781.
22. C. Game and A. Jessop, "Random Coding for Digital Optical Systems", **Proceedings of the 1st European Conference on Optical Communications**, 1975, Conference Publication no. 132, pp 171-173.
23. R. J. Catchpole, "Digital Line Codes and Their Effect on Repeater Spacing", **Proceedings of the Conference on Telecommunication Transmission (IEEE No. 131)**, London, September 9 - 11, 1975, pp 91-94.
24. R. Petrovic, "New Transmission Code for Digital Optical Communications", **Electronics Letters**, Vol. 14, No. 17, August1978, pp 541-542
25. P. A. Franaszek, "Sequence State Coding for Digital Transmission", **BSTJ**, December 1967, pp 143-157.
26. R. Petrovic, "Multilevel Signalling Over Step Index Fibers", **IEEE Transactions on Communications**, Vol. COM-28, No. 2, February 1980, pp 294-297.
27. E. Balicco and G. Iudicello, "Binary Coding in Fiber Optic Digital Transmission With Respect to Timing Extraction and Error Monitoring", **Proceedings of XXIII Congresso Internazionale per l'Elettronica**, Rome, 1976, pp 48-58.

28. D. W. Smith and M. R. Matthews, "Laser Transmitter Design for Optical Fiber Systems", **IEEE Journal on Selected Areas in Communications**, Vol. SAC-1, No. 3, April 1983, pp 515-523.

29. T. L. Maione, D. D. Sell and D. H. Wolaver, "Atlanta Fiber System Experiment: Practical 45 Mb/s Regenerator for Lightwave Transmission", **BSTJ**, Vol. 57, No. 6, July-August 1978, pp 1837-1856.

30. K. Ogawa, "Considerations for Optical Receiver Design", **IEEE Journal on Selected Areas in Communications**, Vol. SAC-1, No. 3, April 1983, pp 524-532.

31. P. O'Neil, "Using Pulse-Bipolar Encoding in Fiber Optic Links", **Electro-Optical Systems Design**, Vol. 13, No. 12, December 1981, pp 33-35.

32. M. Rocks, "Measurements Performed on L.E.D. 4-Level Digital Optical-Fibre System For 68 Mbit/s", **Electronics Letters**, Vol. 15, No. 21, October 11 1979, pp 676-677.

33. H. Grothe et al., "Transmission Experiments and System Calculations Using 1.3μm LED", **Journal of Optical Communications**, Vol. 3, No. 2, 1982, pp 63-66.

34. K. W. Cattermole, "Principles of Digital Line Coding", **International Journal of Electronics**, Vol. 55, No. 1, 1983, pp 3-33.

35. Northern Telecom, "Optical Fiber Digital Transmission Equipment FD-3 Performance Specifications", Practice 321-2111-180, May 1982.

36. Northern Telecom, "Optical Fiber Digital Transmission Equipment FD-2 Description", Practice 321-2101-100, October 1981.

37. Y. Takasaki, M. Tanaka, N. Maeda, K. Yamashita and K. Nagano, "Optical Pulse Formats for Fiber Optic Digital Communications", **IEEE Transactions on Communications**, Vol. COM-24, No. 4, April 1976, pp 404-412.

38. A. F. Graves and F. M. Pascoe, "Quaternary Coding and Potential Applications in the Copper Network", BNR Report WP31286, September 1980.

39. G. Hanke, "Fast 7B-8B Coder And Decoder For An Optical Transmission System", **Proceedings of the IEEE International Conference on Communications**, July 1983, pp c3.4.1-c3.4.5.

40. M. C. Betts et al., "Method of and Apparatus for Code Conversion of Binary to Multilevel Signals", **Canadian Patent** No. 1,153,118, August 30, 1983.

41. M. Rocks, "The Effect of Splices and Equidistant Amplitude Level Spacings in Digital

Multi-Level Optical Waveguide Transmission Systems", **Journal of Optical Communications**, Vol. 2, No. 2, 1981, pp 49-53.

42. J. Dilley, "Small Capacity Digital Line Technical Analysis", BNR Report WP31392, June 1981.
43. M. H. El Diwany, D. J. Roulston, and S. G. Chamberlain, "Design of Low Noise Bipolar Transimpedance Preamplifier for Optical Receivers", **IEEE Proceedings**, Vol. 128, Pt. G, No. 6, December 1981, pp 299-304.
44. C. Kelly, "Multilevel Signalling and Decision-feedback Equalization over Multimode Optical Fibers", M. Sc. thesis, University of Alberta, Fall 1984.
45. W. R. Bennett, "Statistics of Regenerative Digital Transmission", **BSTJ**, Vol. 37, Nov. 1958, pp 1501-1542.
46. G. V. Konovalov and Y. M. Tarasenko, "Matrix Methods for Assessing the Statistical Characteristics of Pulse-Code Signal", **Radio Engineering**, Vol. 25, Feb. 1970, pp 80-86.
47. H. Yasuda and H. Inose, "Direct Calculation Method for Power Spectra of Pulse Sequences by Means of Transition Probability Matrices", **Electronics and Communications in Japan**, Vol. 53-A, Nov. 1970, pp 33-40.
48. H. Yasuda, "Calculation of Power Spectra of Pulse Sequences by Means of Transition Probability Matrices", **Electronics and Communications in Japan**, Vol. 54-A, Nov. 1971, pp 1-7.
49. B. S. Bosik, "The Spectral Density of a Coded Digital Signal", **BSTJ**, Vol. 51, April 1972, pp 921-932.
50. L. J. Greenstein, "Spectrum of a Binary Signal Block Coded for DC Suppression", **BSTJ**, Vol. 53, August 1974, pp 1103-1126.
51. G. L. Cariolaro and G. P. Tronca, "Spectra of Block Coded Digital Signals", **IEEE Transactions on Communications**, Vol. COM-22, No. 10, October 1974, pp 1555-1563.
52. G. L. Cariolaro, G. L. Pierobon, and G. P. Tronca, "Analysis of Codes and Spectra Calculations", **International Journal of Electronics**, Vol. 55, No. 1, 1983, pp 35-79.
53. D. A. Lindholm, "Power Spectra of Channel Codes for Digital Magnetic Recording", **IEEE Transactions on Magnetics**, Vol. MAG-14, No. 5, September 1978, pp 321-323.
54. G. S. Poo, "Computer Aids for Code Spectra Calculations", **IEE Proceedings**, Vol. 128, Pt. F, No. 5, October 1981, pp 323-330.

55. G. Bilardi, R. Padovani, and G. L. Pierobon, "Spectral Analysis of Functional Markov Chains with Applications", **IEEE Transactions on Communications**, Vol. COM-31, No. 7, July 1983, pp 853-860.

56. F. R. Gantmacher, **The Theory of Matrices**, Vol. 1 and 2, Chelsea Publishing Company, New York, 1971, pp 82-98.

57. I. I. Gihman and A. V. Skorohod, **The Theory of Stochastic Processes**, Vol. 1, Springer-Verlag, 1974.